ZX COMPUTING MONTHLY

AN ARGUS SPECIALIST PUBLICATION
MARCH 1987 £1.50

FOR ALL SINCLAIR USERS

MINI STUDIO: TYPE-IN SOUND SAMPLER

FIRST STEPS IN MACHINE CODE

THE DISCIPLE: REVIEWED

PRINTER GRAPHICS



MONSTER HITS:
Silent Service
Gauntlet
BMX Simulator

COMPETITIONS!
Icon Graphix
Cosmic Shock Absorber

IMPOSSABALL

IMPOSSABALL - A BOUNCING IQ TEST?



AMSTRAD 8.95 COMMODORE 8.95 SPECTRUM 7.95
SHORT CIRCUIT™
It was a one-in-a-million accident – but Number Five, designed to be a strategic artificially intelligent weapons system, the most sophisticated robot on the planet, has escaped – and has come to the conclusion that he's alive! Now the scientist who put him together wants to take him apart again to find out what went wrong. The president of Nova Robotics wants to capture him before the weapons he's carrying kill millions of civilians. And the security chief wants to blow him up so that he can get home in time for dinner.
YOU are Number Five...YOU are alive and YOU have got to stay that way!
ocean
Ocean Software Limited
Ocean House · 6 Central Street · Manchester · M2 5NS
Telephone 061 832 6633 · Telex 669977 Oceans G
Short Circuit is a trademark of Tri-Star Pictures, Inc. and PSO Presentations.
© 1986 Tri-Star Pictures, Inc. and PSO Presentations. All Rights Reserved.

CONTENTS

FEATURES



Gauntlet (82)

PROGRAMMING PROJECTS

COMPETITIONS

REGULARS

SOFTWARE REVIEWS



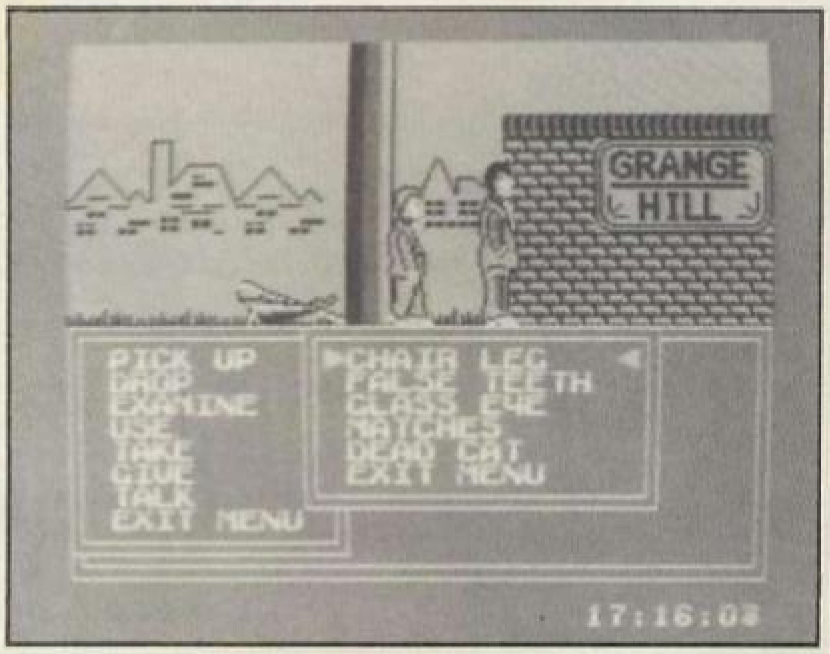
Football Fortunes (12)

N·E·W·S

Grange Hill After Hours

The Grange Hill cast get the arcade adventure treatment in Argus Press Software's release based on the Phil Redmond book, Grange Hill After Hours. The plot in this animated icon driven game centres on Luke "Gonch" Gardener's attempt to retrieve his confiscated Walkman from the school. With his accomplice Paul "Hollo" Holloway he must break into the school after hours, get his Walkman and so avoid the wrath of his mum.

Not surprisingly they run into many of the other characters such as Griffiths the caretaker and Imelda Davies, the school witch and sometimes when things get dicey Hollo has to be pleaded with to carry on. There's a degree of interaction that enables you to talk to characters and their response is in keeping with their character. Grange Hill — The Computer Game costs £9.95.





Monty Returns Again

Gremlin's Monty Mole is back for his fourth outing in Auf Wiedersehn Monty. Peter Harrap, author of the first Monty platform escapade has been recalled to produce what is said to be positively the last Monty game. In the mark IV Monty you must guide him across Europe while in the process picking up sufficient funds to buy the Greek Island of Montos where he can enjoy a peaceful retirement. But will it be the last Monty game? Will Peter Harrap wake up one day soon and find Monty in the shower doing a Bobby Ewing and eager to traipse over the platforms yet again. The answer of course lies in how many Monty fans shell out for the latest last game.

Spectrum Games Top Ten

1 (3)	Gauntlet	US Gold
2 ( )	Footballer of the Year	Gremlin
3 (1)	Trivial Pursuit	Domark
4 (7)	Space Harrier	Elite
5 ( )	Super Soccer	Imagine
6 (5)	The Great Escape	Ocean
7 (2)	Computer Hits 3	Beau Jolly
8 (3)	Paper Boy	Elite
9 ( )	Headcoach	Addictive
10 ( )	Breakthru	US Gold

(Chart supplied by W.H. Smith)

Editor: **Bryan Ralph**
Assistant Editor: **Cliff Joseph**
Consultant Editor: **Ray Elder**
Advertising Manager: **John McGarry**

Design: **Argus Design**
A.S.P. Advertising and Editorial
No. 1 Golden Square, London W1R 3AB 01-437-0626

Printed by Chase Web, Estover, Plymouth.
Advertisement Copy Controler: Andy Selwood
Distributed by: Argus Press Sales and Distribution Ltd, 12-18 Paul Street, London EC2A 4JS

ZX Computing Monthly is published on the fourth Friday of each month. Subscription rates cn be obtained from ZX Subscriptions, Infonet, Times House, 179 The Marlowes, Hemel Hempstead, Herts HP51 1BB.



Joystick To Hand

The makers of the Phasor One joystick are hoping to stun the opposition with their new "handheld" stick design. Made by Britannia Software, the Phasor One can be used by either right or left handed gamers and features the joystick placed on top of the pistol grip with the trigger placed ideally for the index finger. Complete with an extra long cable, presumably for firing on the move, the Phasor One costs £12.95.
Britannia Software (0222 481135)

New Alignments

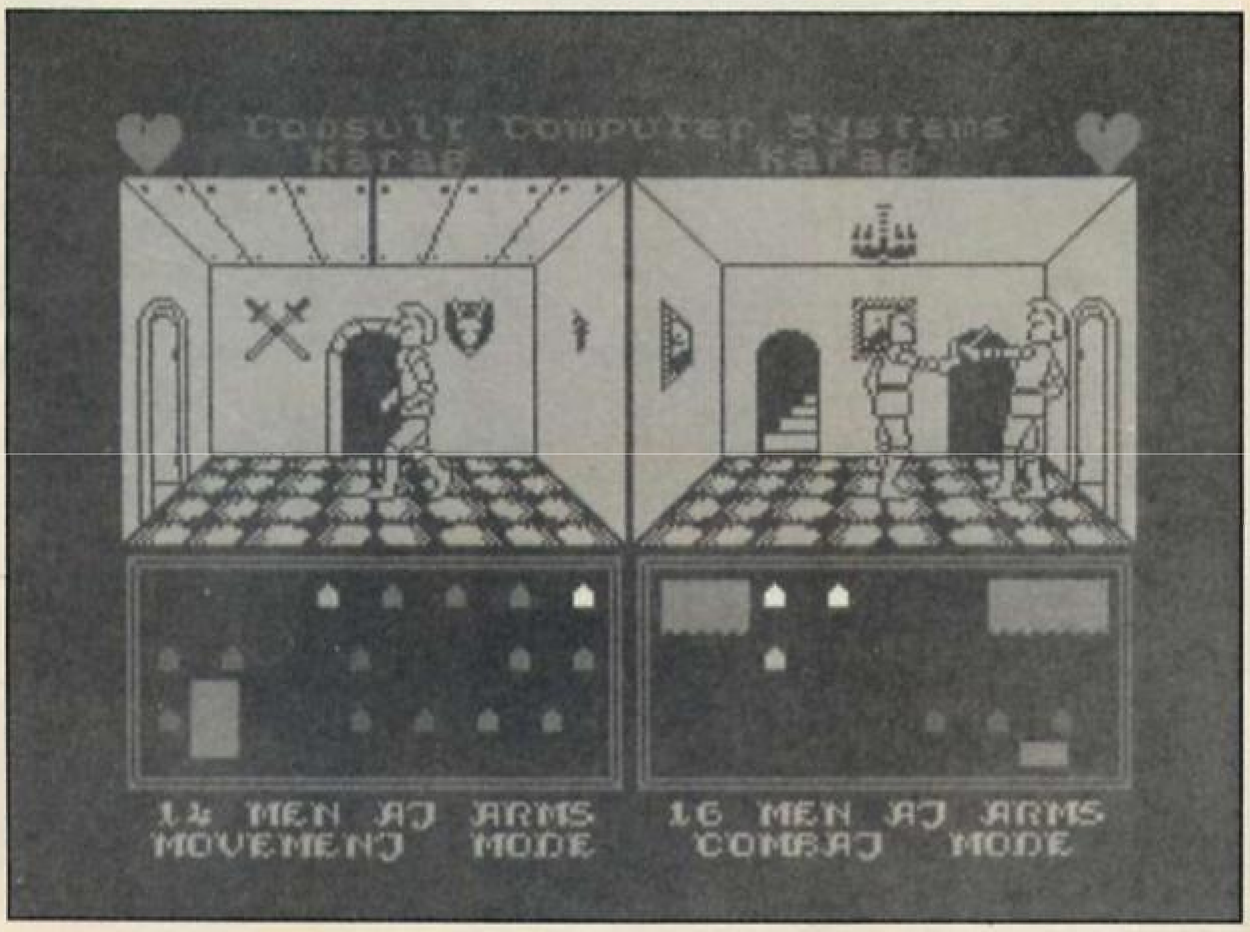
Loading problems? Interceptor Micros have just released an Azimuth Head Alignment Tape for 48K and both 128K Spectrums. Avid Azimuth checkers should note that the package which includes the head alignment program, a test program on the reverse side (Wheelin Wallie), instruction manual, screwdriver and pointer, retails at £8.99.



Yearplanner Winners

A legion of ZX readers set out to rhyme their way to a Level 9 Yearplanner. Perhaps our favourite was Alistair May of Elgin, Scotland who managed to rhyme "adventures" and "dentures" in a poetic gem worthy of William McGonagle.

The other winners are:
R. Wilkinson, Rugby; Jay Sachania, Milton Keynes; C. Grant, Irvine; R. Tarr, Bracknell; G. Galloway, Liverpool; M. Ackroyd, London SE15; D.I. Wicker, Fakenham; J. Jacobsen, Glostrup, Denmark; R. Thrower, London SW18; T. Ridley-Thompson, Lambley; I. Rogers, Devizes; P. Owen, Sidcup; M. Summers, Morpeth; G. Bond, Co Antrim; D.V. Filip, Geraardsbergen, Belgium; C. Collins, Normanto; B. Scott, Gateshead; M. Maysami, Duisburg, W. Germany; H. Williamson, Moray; J. Welsh, Galsgow; T. Jamieson, St Helens; E.J. Chun, Lancaster; J. Shelling, Cardiff; R. Sands, Skipton; P. Laycock, London SE6; M. Kemp, London SE18; H. Spall, London SE7; P.C.B. Page, Liverpool; M. Austin, Chesham; E. Bennet, London SW11; G. Blake, Glasgow; C. Morgan, Croydon; M. Benjamin, Northants; A. Beale, Blandford Forum; M. Beaman, Ilford; P. Reynolds, Sleaford; D. Rademaker, Palinghuizen, Gent, Belgium; D. Morrison, Aberchirder; A. McLellan, Paisley; B. Carter, London E2; A. Rigg, York.



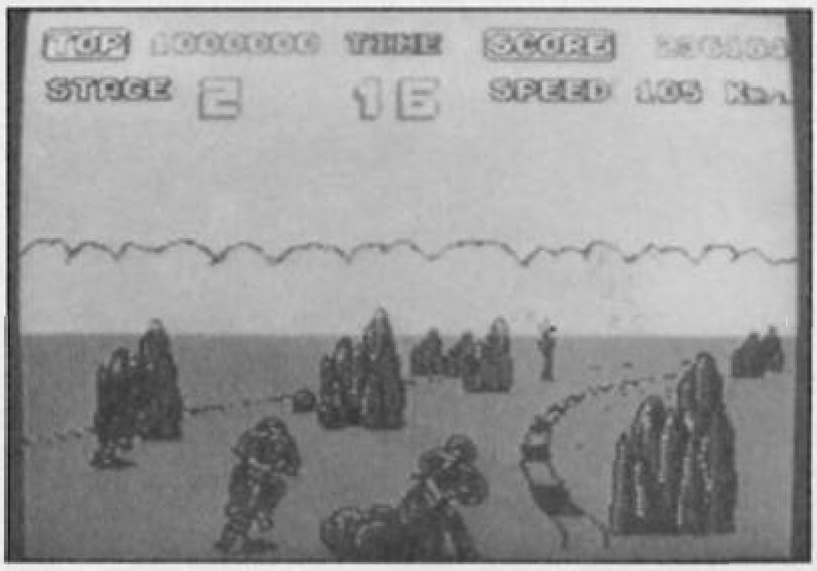
Throne Together

Mike Singleton of Lords of Midnight and Doomsdark Revenge fame is teaming up with Melbourne House to release a "multi-player combat game" called Thrones of Fire. The game is a three-sided power struggle with two players and a computer opponent or one player and two computer adversaries. The three roles played are those of sons of the late King Atherik each vying for the vacant throne. Alliances can be formed (and broken of course) as you try to outmanoeuvre your fraternal foes.

Unlike some medieval strategy games Thrones of Fire promises large animated figures hacking away at each other with axe and broadsword. Priced at £8.95 Throne of Fire has a projected release date of April. Meanwhile another Mike Singleton game Dark Sceptre for Beyond Software has still yet to see the light of day. Will Throne of fire suffer the same fate of repeated delays?

Enduro Racer

Activision are set to release the conversion of Sega's coin-op motorcycle simulation Enduro Racer. The game is a gruelling trail bike marathon over progressively rougher terrain. Enduro Racer will cost £8.95.

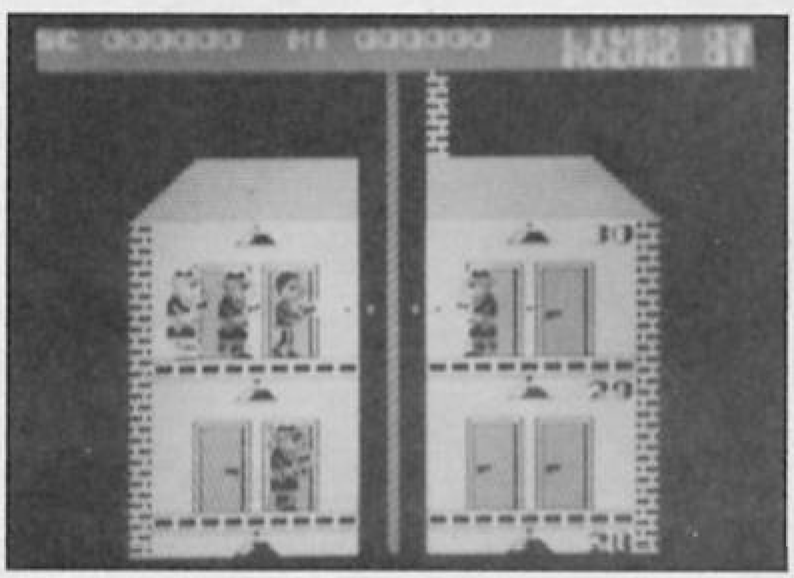


Super Sunday

In the wake of the Superbowl, Nexus are releasing an American Football simulation called Super Sunday. Costing £9.95 Super Sunday will feature 20 of the teams who have appeared in the Superbowl since 1968. Details of the 1987 finalists Denver Broncos and the New York Giants will also be included.

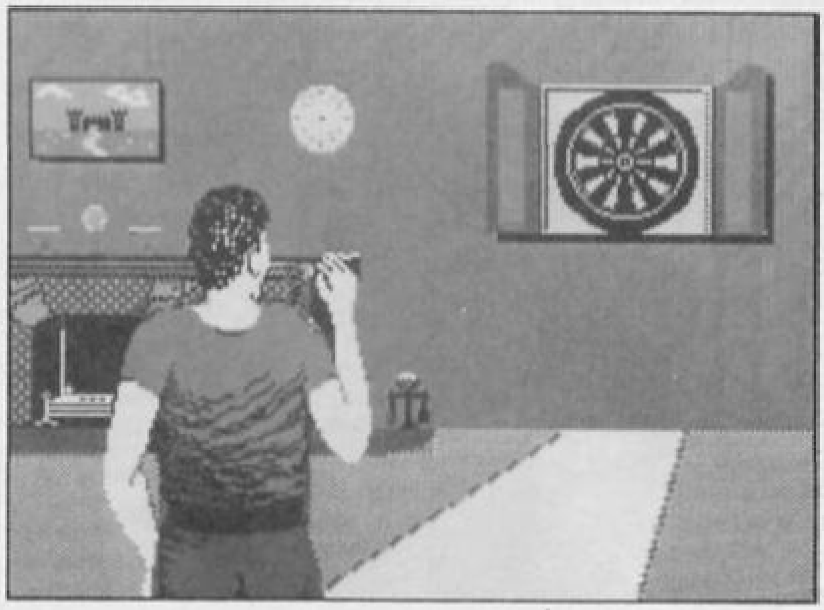
Super Sunday will combine animation with players' performances based on past form statistics. When you've decided which move your team will make, all the players on screen move during the play. There is a two player option or you can take on the computer. There is also an "autoplay" option where the computer plays two sides against the other based purely on the teams' past form.

Lift for Quicksilva

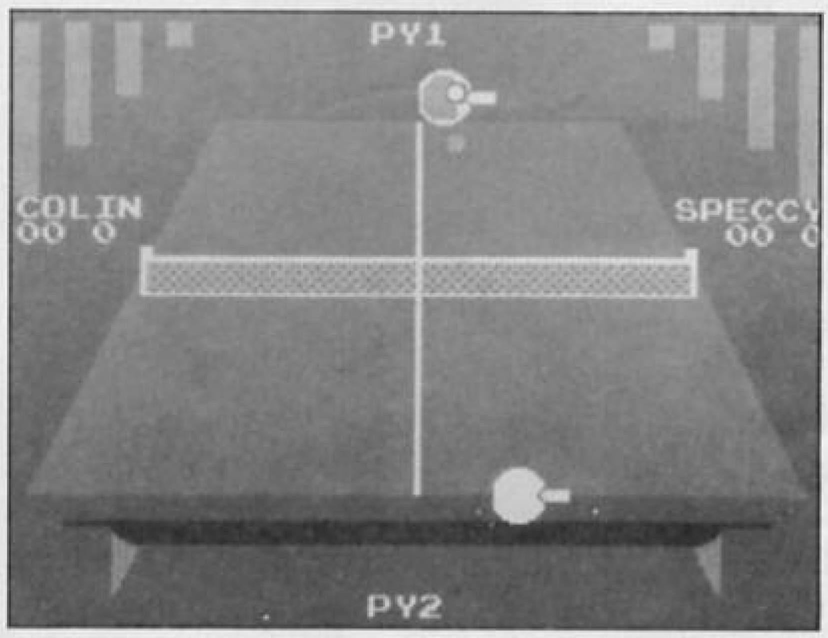


Elevator Action is a game where former lift operators will have a definite advantage. In this coin-op conversion you must guide Agent Otto from the top of a skyscraper to the bottom where his getaway car is waiting. Otto can certainly take care of himself as he is able to "shoot, jump, ride, duck and karate kick his way to safety". The game retails at £8.95.

Indoor Games







Screenshots from Advance Software's four-in-one Indoor games featuring 10 pin bowling, darts, air hockey and ping pong which will retail for £8.95.

Adventures On Paper

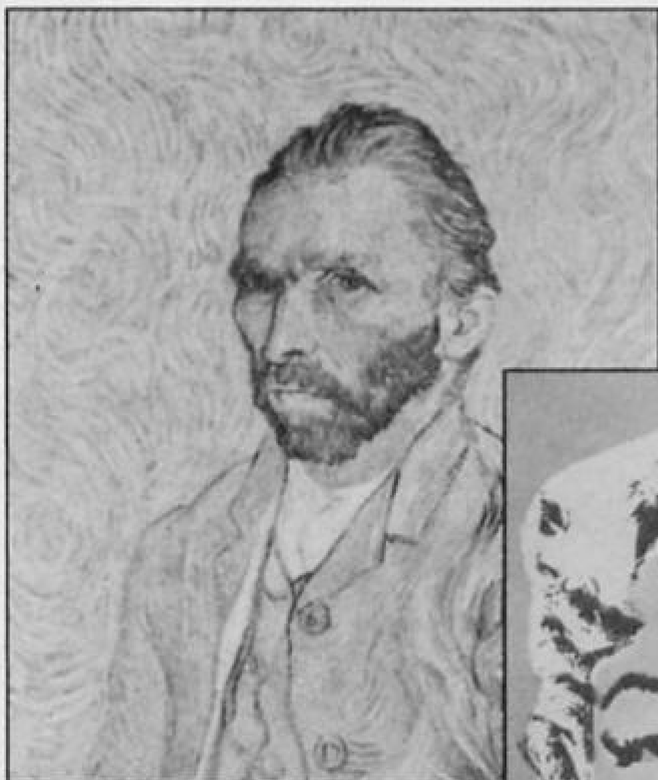
Incentive Software, producers of the Graphic Adventure Creator have released a new Designer pad for Adventurers. Each pad consists of over 200 double sided A4 pages and is aimed at those who want to organise their adventure writing more efficiently. The GAC Designer Pad costs £7.95 plus £1.25 p&p. Incentive Software, (07356 77288).

ICONS AND OLD MASTERS

COMPETITION

Here's your chance to win one of twenty-five copies of Icon Graphix

A



B



C



Copies of Audiogenics' enhanced Icon Graphix for the 128 are on offer in this competition designed for the art connoisseur. If you are a 48K Spectrum owner, don't despair, just tick the appropriate box on the entry coupon and if you win you'll receive a copy of the original Icon Graphix.

All you have to do to win is correctly identify the three famous pictures on this page and name the artists who painted them.

If you can spot all three pictures and their painters and are among the first twenty-five entries pulled out of the hat, you will soon be creating modern masterpieces on your Spectrum with Icon Graphix.

The competition is open to all ZX readers except employees of Argus Specialist Publications, Chase Web and Audiogenic Software Ltd. The editor's decision is final and no correspondence.

Send your entries to Icon Graphix Competition, ZX Computing, No 1 Golden Square, London W1R 3AB.

Please remember to put your address on your entry envelope.

The closing date for entries is April 10th.

Icon Graphix Competition

A) Title — ( ) Artist — ( )

B) Title — ( ) Artist — ( )

C) Title — ( ) Artist — ( )

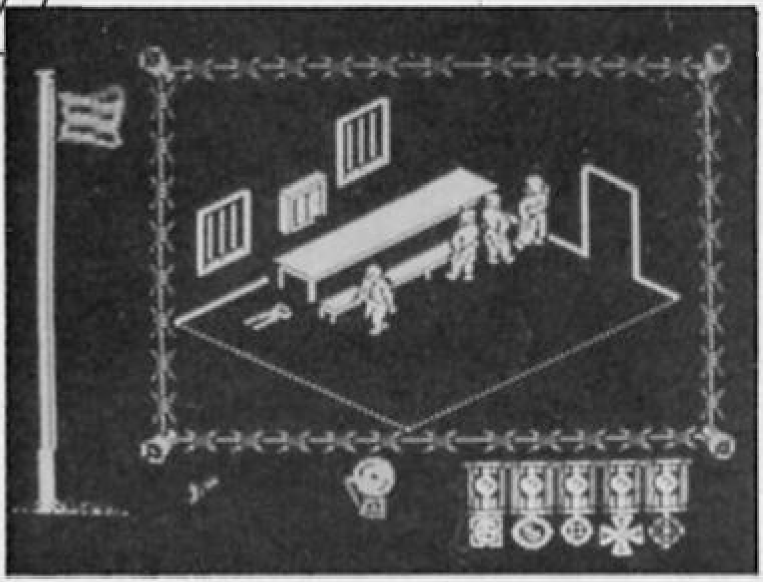
Please tick your machine — 48K ☐ 128K ☐

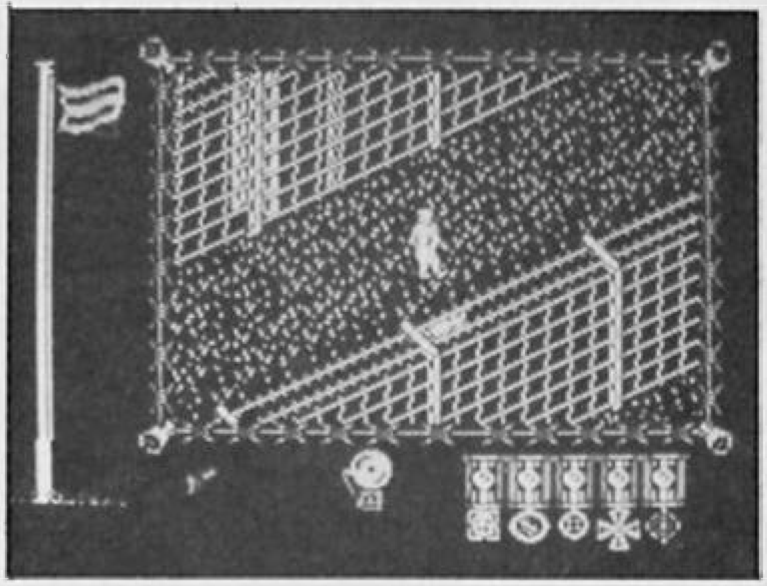
Name ....

Address ....

....

Send your entries to Icon Graphix Competition, ZX Computing, No 1 Golden Square, London W1R3 AB.





A 3D POW escape game that captures the imagination

The Great Escape
Ocean
£7.95

Locked in a prison camp your daily routine consists of roll calls, exercise periods and escape attempts.

If you're indecisive for more than a few seconds you'll automatically follow the daily routine of sleeping, walking and eating. If you want to try something else then a press of the fire button regains control. Now you can explore the camp, search and find potentially useful objects and scout out possible escape routes. But you must return to the camp routine if you want to stay out of solitary.

Swapping in and out of this "auto prisoner" mode is essential if you are to escape as it ensures you get to roll calls, eat and sleep but allows you some time to explore the camps without raising the alarm.

Get caught in the wrong place at the wrong time and you'll be sent to solitary to cool your heels. This also destroys your morale and can also increase the number of guards that are assigned. A constant escort of three guards makes escaping almost impossible.

Your morale is extremely important and illustrated by a flag at the side of the screen.

As escape attempts fail and objects are confiscated your morale drops. The lower it gets the less likely you are able to break out of "auto prisoner" and escape until finally at zero all player control is lost and the game is over.

After recent Ocean disasters such as Miami Vice and Knight Rider I was surprised at how good this game was. It takes a little while to get into it as you just observe the normal prison routine.

Then you can find the opportunities to duck through a door, search the room contents, steal anything interesting and be out before the guards notice you're missing.

You can only carry around two objects at any time and so you have to find a safe place to stash them before the big escape.

A good hiding place and a possible escape route is the tunnel that you can find by moving the boiler in your room! Unfortunately, the tunnel network is complex and difficult to navigate in the dark, but if you found a torch . . .

The game graphics are excellent as the camp is created from 255 scrolling 3D screens full of prisoners, guards, dogs, barbed wire, buildings (most of them off limits) and searchlights.

All night the camp is enclosed in darkness that allows you to explore freely if you can avoid the patrols and the searchlights.

The odds however are against you and you'll probably spend the rest of the night in solitary.

A brilliant idea, superbly implemented, Ocean's best ever Spectrum game!

MONSTER HIT



The answer to the Spectrum-users prayer

the DISCIPLE



AVAILABLE NOW!

the all purpose interface to take your Spectrum to the limit

DISK INTERFACE
- ★ Allows up to 1.6 Mb of storage on 2 drives
- ★ Loads a full 48K program in 3.5 seconds
- ★ Simple Basic Syntax
- ★ Compatible with any standard drive – single or double density, single/double sided, 40/80 track, 3", 3½", 5¼".

SNAPSHOT BUTTON
- ★ Saves any program instantly – at any point you wish and as often as you like.

PRINTER INTERFACE
- ★ Compatible with the whole range of Centronics Parallel printers.
- ★ Uses normal Sinclair commands – LLIST, L PRINT, COPY
- ★ Permits user defined and screen graphics.

DUAL JOYSTICK PORTS
- ★ Sinclair and Kempston compatible.

NETWORKING
- ★ Operating system allows Disciple to act as file server; shared access to printer and disc drive for up to 64 Spectrums
- ★ Interface 1 Compatible



Use the disciple with any Spectrum – even the new Plus Two model.

the DISCIPLE

the ultimate disk, printer, joystick and network controller for the Spectrum **£89.95** (inc VAT)

Designed by Miles Gordon Technology for Rockfort Products (a trading division of Vignesh Ltd)

FABULOUS PACKAGE PRICES
Disciple/Drive from £155.
Disciple/Drive/Printer from £325
For more information call us on 01-203 0191 or visit our London showroom for a demonstration

To: Rockfort Products, 81 Church Road, London NW4 4DP
Please send me POST FREE.................... (Quantity) DISCIPLE Interfaces @ £89.95 each, including VAT & P&P.

I enclose CHEQUE/POSTAL ORDER for £..........

OR debit by ACCESS ☐ or VISA ☐ *(tick as appropriate)*

CARD NO ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

EXPIRY DATE..........

SIGNATURE..........

NAME *(Block Capitals)*..........

ADDRESS..........

.... Postcode..........

Overseas orders and trade enquiries welcome. Telex 946240. Ref. 190-12450


THANATOS
Spectrum in October, Amstrad in November, Commodore 64 in December.
All £9.95
DURELL
Sales dept.,
Castle Lodge, Castle Green,
Taunton, Somerset TA1 4AB
England
Telephone (0823) 54489 & 54029
software getting harder . . . .


DEEP STRIKE
R.R.P. £9.95
Spectrum & Amstrad
available November 20th
Commodore 64
available January 20th
Trade enquiries to Centresoft
on 021 356 3388
Sales dept.,
Castle Lodge, Castle Green,
Taunton, Somerset TA1 4AB
England
Telephone (0823) 54489 & 54029
DURELL
software getting harder . . .

FOOTBALL FORTUNES



Is it a board game or a computer game? Football Fortunes tries to put a boot in both camps.

Brian Clough's Football Fortunes
CDS
£14.95

When popular board games have been adapted for the home computer they have often been a disappointment. Games that can draw people round a table like a magnet tend to lose their appeal when the monitor screen is the focus of attention.

Perhaps with this in mind, CDS have designed a game that keeps the appealing social aspects of board games (ie. it can get very competitive and there are plenty of opportunities to outsmart your opponents) while farming out all the donkey work — rolling the dice, statistics, tables, results etc. to the computer.

Football Fortunes is an enterprising hybrid that has been well thought out to exert maximum hold over the football fanatics imagination. If you've ever followed football you will definitely have put together your "dream team" with all your favourite players. With Football Fortunes you can try to do just that in your quest to carry off the League and Cup double. Success depends on assembling a strong squad from the available player cards, each card represents a well known player and gives his category (defender, midfield, etc.) and a star rating 1 to 5.

If your favourite player has been left out there are blank cards provided so you can include them. The star ratings are bound to spark an argument or two especially if your particular hero is deemed to be only worth one star. The total star ratings for defence and attack are fed into the computer before each match to indicate your team's current form.

The game proceeds much along the lines of a conventional board game. Throw the computerised dice and proceed around the board, which is sizeable (fitting it into the double cassette size box was a masterpiece of origami) to land on squares that may be to your advantage like sponsorship or send you sliding into bankruptcy. The manager's luck square may be good or bad, and result in a windfall or going seriously into the red due to a managerial crisis. "Selection Problems" is a square to beware of as the computer program chooses one of your players randomly who is either injured or retired causing you to frantically rejuggle the team.

Plenty of variety is built into the board game but to stop any chance of it becoming repititious you can conduct transfer deals and haggle over players at any point in the game and if someone is selling off players to avoid bankruptcy the bargaining can get very fierce indeed.

There's an awful lot of enjoyment to be had out of Football Fortunes whether you enjoy football or not. CDS have brought back the human element and the more players you have (from two to five can play) the better the game. For a change the computer program plays a supporting role to the players and Football Fortunes is all the better for that.

MONSTER HIT

PROJECT X

Ray Elder sets the next programming challenge in this interactive series.

Welcome to the second in this occasional series intended to present a problem, discuss means of arriving at a solution and then leave it to you to produce the coding to do the job.

First may I say that I was pleasantly surprised at the good response from the last Project X. Entries are still arriving and I will present the solution which I think is most efficient, original and interesting in the next issue, meanwhile on to this project which is . . .

The Shuffle

Over the last few years I have seen, used and devised several ways of simulating the shuffling of a pack of cards. As a side product the solution to this may also be a means of producing a non repeating sequence of random numbers which may widen the scope of the applications of this exercise.

1. Identify Problem.

We need to be able to select a series of numbers 1-52, which represent the 52 cards in a pack, in a random or mixed up order with no repetition of previously selected numbers/cards.

2. Breakdown of process.

This is fairly straightforward in theory:

a) Set up a representation of a pack of cards.

b) Swap the cards positions in the pack (Jumble up or Shuffle).

3. Possible Approaches.

In practice the implementation offers several ways. The Random number generator will obviously be called on in order to effect the actual shuffle and the most complex part is likely to be the means of ensuring that selections are not repeated.

In the most basic computer card games (and in one or two early commercial games I have seen this happen) a card is simply selected purely at ramdom. The player loses faith in the program when he/she draws two nine of diamonds or five aces!

One approach would be to select a card at random and then check it against a list of the cards previously chosen to make sure that it hasn't been picked before. This means that all previously selected cards have to be kept recorded to enable the check to be made.

A decision has to be made of the way the card will be selected. It could be a straightforward random number between 1 and 52 (INT (RND*52+1)) and then use a table to look up the actual card and value. IE. 1 may equal the ACE of SPADES, 2 the TWO of SPADES etc. up to 52 perhaps being the KING of DIAMONDS.

If this method is used then each selected number needs to be stored somehow, possibilities are to store as a number or as a string. If kept in numberical form then perhaps the simplest way would be to DIM an array variable to 52 — DIM S(52) — this also initialises each of the 52 elements to 0, and as each number is chosen then change that element to 1 as a flag that it has already been picked.

The check code would be something like IF S(number) THEN GO TO choose again line.

This is perfectly OK for games where only a few cards are required each hand (such as pontoon) but if a lot of cards are used then the selection procedure slows to an unbearable level as many reselections and checks need to be made. You also have the problem of keeping another record of the numbers for reference.

An Alternative?

Rather than fill the empty array it could be set to the numbers first then shift each number around, this also has the advantage of keeping a recording of each number.

Something like **DIM S(52):FOR I=1 TO 52:LET S(I)=I:NEXT I** to set up the array and then select two RND numbers between 1 & 52 and swap the contents of the array. eg: **LET A=INT(RND*52+1):LET B=INT(RND*52+1):LET X=S(A):LET S(A)=S(B):LET S(B)=X**

Put this in a loop for as many times as it takes to get a good mix and away we go!

Of course we still need a table so that we can work out which number represents which card, but the actual shuffle takes a fraction of the time that the first method would take. Personally I wouldn't use either method, but I thought it worth looking at the first two ways because the second method has a wider application than for our card shuffle in that it is a very effective way of producing a numerical non repeating sequence of random numbers.

If you think of a pack of cards then there are two elements, the number or value and the suit — Hearts, Spades, Diamonds and Clubs — the non numerical cards are Ace, Jack, Queen and King. It is almost as if the designer of cards had us computer bods in mind. NONE of these names begin with the same letter!

This means we could set up a string to represent the cards using H,S,D,C,A,J,Q, and K to represent each of these words. A typical set up might be **LET C£="AH2H3H4ZH5H6H7H9HTHJ HQHKHAS2S3S4S5S6S7S8S9STSJ SQSKSAD2D3D4D5D6D7D8D9DT DJDQDKDAC2C3C4C5C6C7C8 C9CTCJCQCKC"**

Note that I used T to represent 10 because this then means that each card is represented by only two characters. Shuffling becomes a matter of picking an element of C$ at random, removing it from the string and adding it to the front or the end of the string for as many times as it takes to get a good mix.

If you set up the variables T,J,Q,K and A to a value of 10 then finding the numerical value of the card is a matter of getting the VAL of the first of the two elements that represent a card. So now what? I hope this has started a few ideas buzzing around and, just like the last time, we would like to see the results of your ideas, especially if you know of, or create a completely unique method of solving the problem.

What we would like is a **short** program which shuffles a pack of cards and deals **two** cards face up. We do not want a full game, but we will take into consideration the solution to the shuffle problem and the quality of the graphic representation of the cards.

The program(s) which impress us most will be featured, and their authors suitably rewarded, in a future issue.

FIRST STEPS INTO MACHINE CODE

Alan Davis makes the transition from BASIC into machine code.

Most people seem to find no great difficulty in becoming reasonable proficient at BASIC programming, after a little practice and experience. But sooner or later they realise the limitations of BASIC, seek to go further — and find themselves up against the notorious "machine code barrier". It's a barrier that, sadly, many fail to cross (though often not for want of trying), and I gather that the question "How do I pass from BASIC to machine code programming?" is still one of the most common requests in the ZXC mailbag. So here's my own attempt to provide a very simple approach with a minimum of jargon — by which I mean VERY SIMPLE INDEED! I'm going to sacrifice discussion of a number of issues which would be theoretically desirable, because the most important thing is to make that initial transition and actually write machine code programs, no matter how trivial. Even if you've tried before and failed, I hope I can persuade you to have another go — because I'm convinced that most people give up simply because they've tried to swallow too much, too soon.

The cursed hex

If have tried to come to grips with machine code before, you've probably been harrassed by hexadecimal, tousled by "two's complement", and stultified by "the stack". But though you'll EVENTUALLY need to get to grips with such things if you're to become really proficient, you can in fact get started without them. There are certain disadvantages in doing so — but we're desperate folk, and desperate measures are called for! So let's begin. What, exactly, is "machine code"?

Though you may not be aware of it, you're using machine code every time you run a BASIC program on your Spectrum. You may think that the computer understands BASIC commands such as PRINT AT 10,12;"Hello!" — but it doesn't. When we program in BASIC we're being spoon-fed by a complex and length program which is present in the Spectrum when we switch on, part of whose task is to translate our BASIC instructions into a form that the heart of the computer — the Z80 microprocessor — can cope with. What the Z80 understands is the stuff we call machine code: a sequence of numbers which cause the Z80 to perform certain actions. If we by-pass the Spectrum's "translating" system and feed the Z80 directly with the correct numbers, we can exercise total control over the machine. Heady stuff!

First steps

So let's write a machine code program. First of all we need to secure an area of memory so that any fiddling about in BASIC that we do won't interfere with it, and we can do this using **CLEAR 59999.** This means that we can safely manipulate any address from 60000 upwards. Next, enter these two direct commands, one after the other:

POKE 60000,0
POKE 6000,201

What you've just done is to store two numbers (which together make up a machine code program) in the Spectrum's memory at addresses 60000 and 60001. What we do now is to direct the Z80's attention to this sequence of numbers, and you can do this by entering RANDOMIZE USR 60000. Try it yourself, now. You might think that nothing has happened — but you'd be wrong; quite a lot has happened, infact. First, the Z80 looked at the contents of address 60000, and found the number 0. This is the machine code "NO OPERATION" instruction which tells the Z80 to do nothing at all. So for a tiny fraction of a second it did indeed do nothing. Then it looked at the contents of the next address (60001) and found the number 201. This is the "RETURN" instruction, which in this case tells the Z80 to leave the machine code program at this point and effectively "return to BASIC". And so the obedient Z80, without further ado, did just that — which is why at the end you were presented with the O.K. report at the bottom of the screen. The machine code program has been completed, and normal service has been resumed.

Now, the good news is that in understanding that, you've understood a machine code program. Of course you'll very rightly point out that it's not a very useful program — but that doesn't matter. The important thing to appreciate is that what to us looks like a string of numbers is effectively a sequence of commands to the Z80. Herein, of course, lies a problem; I'm not a Z80, and neither are you. The number 0 doesn't immediately make us think "DO NOTHING", and the number 201 certainly doesn't make us think "RETURN". If we had to learn to read and write machine code like this, life would be very hard indeed — but fortunately, we don't have to.

What we need, really, is some sort of intermediate language to act as a buffer between ourselves and the Z80, and fortunately such a language does exist. It's called ASSEMBLY language, which is a kind of drastically abbreviated pidgin English where each machine code instruction (that is, a number or series of numbers) is represented by a *mnemonic* (a set of letters) which is much easier for us to understand. To take an example, if we translate our little machine code program above into assembly language, it looks like this:

NOP (No Operation)
RET (Return to BASIC)

This is much easier to get to grips with, isn't it? You can understand it; I can understand it; but the bad news is that the Z80 can't! It wants nothing more nor less than NUMBERS. The solution lies in an ASSEMBLER program — and this, I'm afraid, is the point at which you'll need to blow the dust off your cheque-book. There are plenty of assembler programs on the market and I'm by no means familiar with all of them myself, having settled long ago on Hisoft's DEVPAC assembler. I've also heard of friendlier programs available these days, more suited to beginners.

LISTING 1

Do nothing, then return to BASIC.

```
60000   10          ORG 60000
60000   20          NOP
60001   30          RET
```

LISTING 2

Copy contents of one address into another.

```
60000   10          ORG 60000
60000   20          LD A,(ADDR1)
60003   30          LD (ADDR2),A
60006   40          RET
60007   50 ADDR1    DEFB 2
60008   60 ADDR2    DEFB 0
```

LISTING 3

Read contents of one address, add one, and put result into another address.

```
60000   10          ORG 60000
60000   20          LD A,(ADDR1)
60003   30          INC A
60004   40          LD (ADDR2),A
60007   50          RET
60008   60 ADDR1    DEFB 2
60009   70 ADDR2    DEFB 0
```

LISTING 4

Add the contents of two addresses, and store result in a third address.

```
60000   10          ORG 60000
60000   20          LD A,(ADDR1)
60003   30          LD B,A
60004   40          LD A,(ADDR1)
60007   50          ADD A,B
60008   60          LD (ADDR3),A
60011   70          RET
60012   80 ADDR1    DEFB 2
60013   90 ADDR2    DEFB 3
60014  100 ADDR3    DEFB 0
```

Assembler

A good assembler program, once running, will take over the control of your Spectrum to allow you to enter and edit a program in assembly language, and will then "assemble", or translate, your program into machine code. But it should also do much more. For example, it should allow you to use line numbers which, though not essential, help you to keep track of what you're doing. It will allow you to attach LABELS to certain addresses — which is yet another move towards making your program more closely resemble English. And it will also allow you to use certain other instructions (*assembler directives* and *assembler commands*) which are not translated into machine code, but which instruct the assembler program to take specific actions — and these generally make life easier for you. There's no short cut here, I'm afraid. You'll need to spend a bit of time reading and absorbing the relevant sections of the manual for your assembler in order to discover just what editing facilities are available to you. However, an example will make things clearer — so let's take a look at Listing 1, which is what I got after assembling our "do nothing and return to BASIC" program.

First, I should point out that in order to get all the listings given here I instructed the assembler to print addresses (these are in the left hand column of the listings) in decimal, rather than the more usual hexadecimal — and incidentally, the facility to have this option is something you may like to check when choosing your assembler, it can make things quite a bit simpler. I also instructed it not to bother listing the machine code that it produces; the Z80 needs those numbers, but we don't at this stage. On, then to the program proper. You'll see that line 10 contains an assembler directive which all assemblers will recognise, namely **ORG 60000.** The **ORG** directive is important because it tells the assembler where to put the machine code it generates — and in our case this is to be from addresses 60000 onwards.

Line 20 contains the first actual assembly language instruction (**NOP**), and line 30 completes the program with the RET instruction. The left hand column of the listing is produced automatically when the program is assembled (ie. when the assembler is instructed to translate the mnemonics to actual machine code) and it shows the addresses at which each machine code instruction has been stored — in this case our machine code program uses only two addresses: 60000 and 60001.

Registers

Now this is all very well, but you could reasonably be asking by this time "Can we have a program which actually DOES something, please?" So let's move on and see what else the Z80 can do. You probably know already that the Z80 has a number of REGISTERS, and you may also know that these have names: there's the "A" register, "B" register, "C", and so on. Each of these registers can at any time hold any one number between 0 and 255 (ie. one BYTE — just like any address in the Spectrum's memory). And there are dozens of machine code instructions which allow the Z80 to copy a number from one register into another, and also to communicate with the outside world (ie. the Spectrum's memory). The assembly language instruction LD, B,A, for example, means "copy (or LoaD) the contents of the A register into the B register". Similarly, LD C,B means "copy the contents of the B register into the C register". There are also instructions for putting numbers directly into registers — so that LD A,23 for example means "put the number 23 into the A register".

As you can imagine, we could write a cheerful little

program causing the Z80 to mutter away merrily to itself, swapping numbers back and forth between its registers without affecting the outside world at all — but I don't think that's what you mean by a program that "does something", so let's move hastily on. How can we pass information between the Z80 and the Spectrum's memory?

The simplest way of doing it is by means of an instruction such as LD A, (address). This would cause the Z80 to look at the specified address in the Spectrum's memory, and copy its contents into the A register. You can achieve the reverse process with the instruction LD (address),A — which would make the Z80 copy the contents of the A register into the specified address in the Spectrum's memory. (There's a limit to what you can do in this line, however, and the A register is the only single register which can communicate directly with specified addresses in this way. LD B, (address) isn't allowed, for instance).

I think you'll see that we could use these instructions to copy a byte from one memory location to another, thus:

```
LD A,(60010)
LD (60011),A
RET
```

This would load the number in address 60010 into the A register, copy it into address 60011, and then return to BASIC. (To do the same thing in BASIC, you'd use the instruction **POKE 60011, PEEK 60010**). This would be fine if you knew in advance which memory locations to specify while writing your program, but in fact more often that not, you don't — at least, not until you've assembled your code. It's time for the trusty assembler program to come to the rescue again, so let's have a look at Listing 2.

Labels

There are two new concepts involved in Listing 2, both concerned with the ability of an assembler program to give you an easy ride. The first is the concept of labels. When I typed this assembly language program in, I decided to call my two addresses ADDR1 and ADDR2 respectively — and so lines 20 and 30 contain the instructions needed for copying the contents of ADDR1 into ADDR2. The assembler is perfectly happy about this provided I tell it what I mean by those labels somewhere — and I've done this in lines 50 and 60 using an assembler directive "DEFB number". (Think of DEFB as meaning "DEFine Byte".) When the assembler encounters the DEFB directive it reserves a single byte of the Spectrum's memory and inserts whatever number has been assigned — so in this case it will store the number 2 in ADDR1 and the number 0 in ADDR2. As you can see from Listing 2, the assembler has allocated ADDR1 the value 60007, and ADDR2 the value 6008, neatly tucking our reserved bytes away at the end of the machine code program itself. Incidentally, you may have noticed that the addresses in the left hand column increase initially in steps of three; this is because the machine code instruction corresponding to LD A,(address) consists of three numbers — one to specify the actual instruction LD A,(number) and two more to specify the address concerned. But don't worry about it for the present — that's the assembler's job, not yours.

You can try this for yourself by typing the program into your assembler, assembling it, then returning to BASIC. Before running the machine code, check for yourself that **PEEK 60007** and **PEEK 60008** give the values 2 and 0 respectively. Now enter **RANDOMIZE USR 60000,** and do your PEEKs again. This time BOTH addresses should contain the number 2, providing that your machine code program has indeed copied the contents of the first address into the second. **POKE** another number into 60007, try **RANDOMIZE USR 60000** again, and then have another **PEEK.**

Again, it's important that you do try these things for yourself, as opposed to merely reading about them, if only to become familiar with the use of your assembler program. Try extending the program in Listing so that it will copy the contents of ADDR1 into TWO other addresses (you'll need to add another instruction LD (ADDR3),A, and use DEFB to reserve another byte labelled ADDR3).

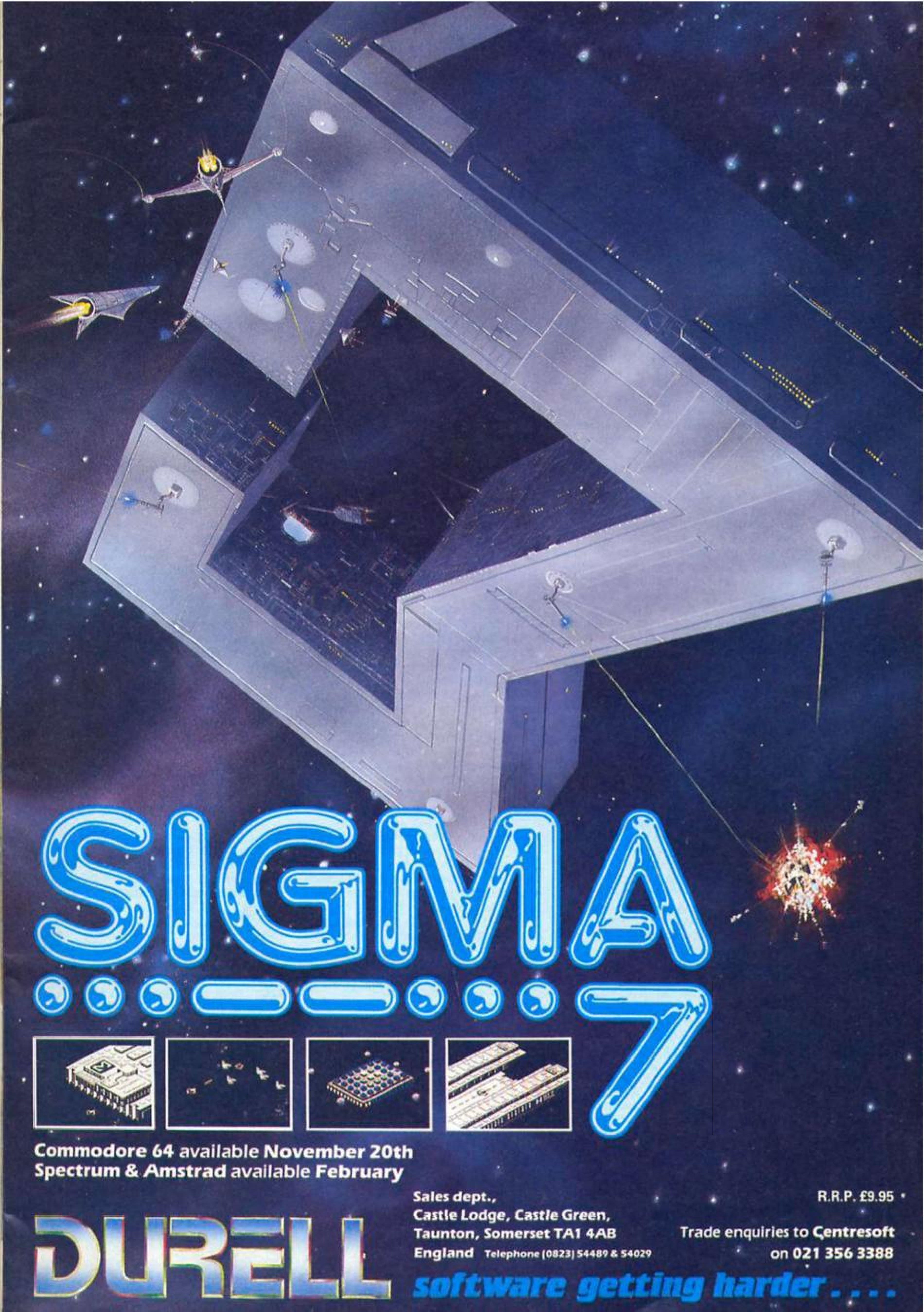
When you've mastered the foregoing, you'll want to stretch your newly grown machine code wings a little further. Instead of merely copying numbers from one place to another, let's see if we can't ring a few changes. Actually, it's not difficult to do so, because simple instructions exist for increasing or decreasing the contents of registers by 1 — for example, INC A ("increase contents of A register by 1") and DEC A ("decrease contents of A register by 1"). Listing 3 shows how INC A can be used in a program to transfer the number held in one memory address to another, increasing it by one en route. Try it for yourself, PEEKing the appropriate addresses both before and after the machine code has been run to check that all is well, and then try writing your own amendments using DEC A — or try using INC A twice in succession to increase a number by 2.

Adding

Our last example — listing 4 — shows how simple addition of small numbers can be achieved using the instruction ADD, A,B ("add contents of B register to contents of A register, leaving the result in A"). If you follow this routine through, you'll see that the contents of the memory address ADDR1 are copied directly into the A register, and then into the B register using LD B,A. (We need to take this indirect route because there's no LD B,(address) instruction). Then A reads in the contents of ADDR2, and the addition is performed by ADD A,B. The result is now in the A register, so we finished off by copying it into a third address, ADDR3 before returning to BASIC. Again, you can check it all for yourself by PEEKing ADDR1 (60012), ADDR2 (60013), and ADDR3 (60014) before and after running the machine code. You can also try different numbers, of course, by poking them into ADDR1 and ADDR2 from BASIC. If you want to try subtraction, then by all means do so: the instruction you need is SUB B ("subtract contents of B register from contents of A register, leaving the result in "A") — but for the present you'd do well to ensure that you use numbers which will produce a positive result.

This is as far as I intend to go in this article, and you may be understandably disappointed that our progress has been decidedly limited. Well yes, it has — and very deliberately so. I know that you want to learn about all the other registers; to be able to use programming loops; to handle numbers in excess of 255; and lots more besides. That's a job for the future, though, and the important thing for now is to get your foot on the first rung of the machine code ladder; to make that critical transition towards thinking in terms of registers and addresses. Use the ideas in this article to write simple routines for yourself — no matter how useless they may be except as practice examples — and then congratulate yourself when you get them working. Familiarity is the key.

Once you've got this far, then you're on your way. But what next? Well, you could do much worse that to buy "Machine Code for Beginners" published by Usborne at a mere £1.99. This will gently fill in much background material that I've omitted here, and indeed will take you further on. Or, you could wait until next month when I'll be delving further into machine code . . .


SIGMA 7
Commodore 64 available November 20th
Spectrum & Amstrad available February
Sales dept.,
Castle Lodge, Castle Green,
Taunton, Somerset TA1 4AB
England Telephone (0823) 54489 & 54029
R.R.P. £9.95
Trade enquiries to Centresoft
on 021 356 3388
DURELL
software getting harder . . . .

EXPERT SYSTEMS

Transforming your Spectrum or QL into an oracle of advice — with David Nowotnik.

Table 1. Main Elements of an Expert System

1. Knowledge Base
2. Inference Engine
3. User Interface
4. Use of Statistics and Probabilities
5. Ability to learn new rules/knowledge
6. Provide an explanation of its decision

Last month, in part 1 of this series, I described the essential elements that go to make a computer 'expert system'. As a reminder, an expert system provides the means to place into a computer not only the knowledge but the experience of a human expert, which a non-expert can interrogate to obtain benefits such as advise. **Table 1** lists the important parts of an expert system, and in this article, and in the next few months I'll be examining some of these elements, providing demonstration programs for QL and Spectrum computers.

Like any other programming problem, there is no single solution to the production of an expert system. There are many algorithms already, and many more ideas are being developed, so all I can give you in the next few months is a picture of what makes an expert system. This month we shall examine two ways in which you can get a computer to come to a decision.

Go along to a human expert for advise, and the usual course of events is something like this:

First of all the human expert will ask you to define the problem, to determine exactly what you wish to know from him. Next he will ask you questions. These questions are selected by the expert to provide him with information, which he can match against his own knowledge base. It is experience which guides the direction of his questions. In an ideal world, he will end up with sufficient information to provide a perfect match with something in his knowledge base to give you a single, unequivocal answer.

The real world isn't quite like that; your expert's knowledge base may not be perfect for the problem you have. You may not be able to provide all the information required. You are more likely to end up with one or more opinions from your expert, each opinion with different level of certainty.

Lack of certainty (uncertainty) is something I'll consider later in the series; for now, let's consider ourselves and our computer expert to be perfect!

The majority of computer expert systems produced to date are designed to tackle a fairly specific problem. So, with the computer 'expert', you have already defined the problem you wish to have solved by selecting a particular expert system. When you RUN any expert system, you should get a message to tell you what areas of expertise that program is capable of dealing with, and you will immediately start the question and answer dialogue.

How does the expert know which questions to ask, and what does he do with the information he receives? These are two important questions in designing an expert system. In the rest of this article, I'll be considering two solutions.

Consider this rather trivial problem. You want your expert to tell you the name of a particular mode of transport. The expert will ask you a series of questions relating to features of the type of vehicle you have in mind, and after a number of questions and answers, he is able to name the type of transport.

Yes, it is trivial, but I have deliberately chosen an example in which we should all be 'experts', so that you can understand the logic of choosing questions, and the interpretation of answers.

In the example programs I provide, our computer 'expert' is aware of 13 modes of transport (see program lines 1500 to 1630 in Fig. 1 or Fig. 4). One strategy for finding the right answer is to ask a series of questions, each answer in turn able to eliminate a group of vehicles, until only one is left. This strategy is called serial decision making.

Pathways

For this example, the human expert on transport (me!) has selected a set of questions to which the user provides yes/no answers. There is a set path through a series of questions from the start of the interrogation to any one particular answer. **Fig. 2** shows all the paths through the question and answers in the expert system listing given in **Fig. 1.**

The first question is always the same; does the vehicle transport people on land? A 'yes' answer immediately eliminates water craft and aircraft; the 'expert' then asks if it travels on rails. If the answer is 'yes', the problem is solved, and the answer 'train' is given; if the reply is 'no', then the expert knows the vehicle travels on road, and two more questions are required to determine which of the four road vehicles known to our 'expert' is the one in question.

For any expert system using this serial approach to decision making, the human expert designing the system has to be aware of all possibilities. The questions have to be unambiguous, so that the user can provide the correct answer every time; any one incorrect response always ends in a wrong answer from our expert.

But, before we examine the failings of this approach too deeply, why not try it out for yourself. The listing in Fig. 1 is suitable for both the QL and Spectrum. The program was written on the QL (hence the command REMark, which will appear as REM on the Spectrum), but tested on both machines.

QL SuperBASIC perfectionists will be horrified by all the GOTOs and GOSUBs, but some allowance had to be made for the Spectrum's less structured BASIC! Spectrum users should use '+' instead of '&' for joining strings (lines 5200 and 5210), and

Figure 1: QL/Spectrum Listing

```
10 REMark Simple Expert System
20 REMark Using serial decision-making
30 REMark
40 REMark David Nowotnik
50 REMark January, 1987
60 REMark
100 REMark Initialise & Title
110 LET x=2000
120 RESTORE : READ t$
130 CLS: PRINT t$
140 PRINT: PRINT
150 PRINT "I will ask you a series of"
160 PRINT "questions. Answer with y"
170 PRINT "for yes or n for no for"
180 PRINT "each one until I provide"
190 PRINT "you with the solution."
200 PRINT : PRINT
210 REMark
220 REMark Main Loop
230 RESTORE x
240 READ a,b,c
250 IF b=0 THEN GO TO 5000
260 GO SUB 5140
270 INPUT z$: IF z$="" THEN GO TO 270
280 IF z$(1)="y" OR z$(1)="Y" THEN LET x=b: PRINT "
yes": GO TO 230
290 IF z$(1)="n" OR z$(1)="N" THEN LET x=c: PRINT "
no": GO TO 230
300 GO TO 270
310 REMark
630 REMark
800 DATA "DETERMINE THE TYPE OF TRANSPORT"
810 REMark
900 REMark Questions
1000 DATA "Does it travel on land"
1010 DATA "Does it travel on rails"
1020 DATA "Does it have two wheels"
1030 DATA "Is it powered by an engine"
1040 DATA "Can it carry more than 6 passengers"
1050 DATA "Does it travel more than 20 feet above
the surface"
1060 DATA "It travels on water; does it only float
on the surface"
1070 DATA "Does it have fixed wings"
1080 DATA "Is it wind powered"
1090 DATA "Can it travel below the surface"
1500 REMark Answers
1510 DATA "train"
1520 DATA "motor bike"
1530 DATA "bicycle"
1540 DATA "coach"
1550 DATA "car"
1560 DATA "aeroplane"
1570 DATA "helicopter"
1580 DATA "glider"
1590 DATA "balloon"
1600 DATA "yacht"
1610 DATA "ship"
1620 DATA "submarine"
1630 DATA "hovercraft"
1890 REMark
1900 REMark Knowledge Base
2000 DATA 1000,2010,2050
2010 DATA 1010,3000,2020
2020 DATA 1020,2030,2040
2030 DATA 1030,3010,3020
2040 DATA 1040,3030,3040
2050 DATA 1050,2060,2090
2060 DATA 1030,2070,2080
2070 DATA 1070,3050,3060
2080 DATA 1070,3070,3080
2090 DATA 1060,2100,2110
2100 DATA 1080,3090,3100
2110 DATA 1090,3110,3120
2900 REMark
2990 REMark Answers
3000 DATA 1510,0,0
3010 DATA 1520,0,0
3020 DATA 1530,0,0
3030 DATA 1540,0,0
3040 DATA 1550,0,0
3050 DATA 1560,0,0
3060 DATA 1570,0,0
3070 DATA 1580,0,0
3080 DATA 1590,0,0
3090 DATA 1600,0,0
3100 DATA 1610,0,0
3110 DATA 1620,0,0
3120 DATA 1630,0,0
5000 REMark Provide the answer
5010 PRINT
5020 RESTORE a: READ e$
5030 PRINT "The answer is a";
5040 IF e$(1)="a" OR e$(1)="e" OR e$(1)="i" OR
e$(1)="o" OR e$(1)="u" THEN PRINT "n ";: GO TO 5060
5050 PRINT " ";
5060 PRINT e$
5070 PRINT: PRINT "Any more? (y/n) ";
5080 INPUT z$
5090 IF z$="" THEN GO TO 5080
5100 IF z$(1)="y" OR z$(1)="Y" THEN RUN
5110 IF z$(1)="n" OR z$(1)="N" THEN STOP
5120 GO TO 5070
5130 REMark
5140 REMark Print question string
5150 RESTORE a: READ e$
5160 LET i=0: LET LE=LEN(e$): LET LI=0
5170 LET x$=""
5180 LET i=i+1: IF i>LE THEN GO TO 5210
5190 IF e$(i)=" " THEN GO TO 5210
5200 LET x$=x$&e$(i): GO TO 5180
5210 IF i>LE THEN LET x$=x$&"?"
5220 IF LI+LEN (x$)>31 THEN PRINT: LET LI=0
5230 PRINT x$;: LET LI=LI+LEN (x$)
5240 IF LI<31 THEN PRINT " ";: LET LI=LI+1
5250 IF i<LE THEN GO TO 5170
5260 RETurn
```

Figure 2: Decision Tree for the transport serial decision making example



for QL users, the commands in italics in lines 280 and 290 are optional.

The DATA statement in line 800 provides the title and purpose of our expert system. Once this is printed (line 130) and general instructions appear on the screen (lines 150 to 190), the main question and answer loop is entered. The questions, answers, and 'knowledge base' all appear in DATA lines, and RESTORE is used extensively to retrieve the appropriate piece of information.

On the first circuit of the main loop, the variable x is set to 2000, which is the DATA line for the first question. From this DATA line, three numeric variables are READ, a,b, and c. 'a' is the line number of the question (character string) to be asked, 'b' is the appropriate DATA line in the knowledge base if the

Figure 3: Alternative knowledge base for Serial Demonstration Program

```
800 DATA "AXIAL AND APPENDICULAR BONES"
810 REMark
900 REMark Questions
1000 DATA "Is this bone in the hand/arm"
1010 DATA "Is this bone in the arm"
1020 DATA "Is this bone in the upper arm"
1030 DATA "There are two bones in the lower arm; is
it the one nearest the body with palms forward"
1040 DATA "Is this bone in the fingers"
1050 DATA "Is this bone at the end of the fingers"
1060 DATA "Is this bone in the legs/feet"
1070 DATA "Is this bone in the ankles/feet"
1080 DATA "Is this bone in the ankle"
1090 DATA "Is this bone in the upper leg"
1100 DATA "There are two bones in the lower leg; is
it the outermost one"
1110 DATA "Is this bone in the shoulders"
1120 DATA "Is this bone attached to the humerous"
1500 REMark Answers
1510 DATA "humerous"
1520 DATA "ulna"
1530 DATA "radius"
1540 DATA "phalanges"
1550 DATA "metacarpels"
1560 DATA "carpels (wrist)"
1570 DATA "tarsels"
1580 DATA "metatarsels"
1590 DATA "femur"
1600 DATA "fibula"
1610 DATA "tibia"
1620 DATA "scapula"
1630 DATA "clavicle"
1640 DATA "sternum"
1890 REMark
1900 REMark Knowledge Base
2000 DATA 1000,2010,2060
2010 DATA 1010,2020,2040
2020 DATA 1020,3000,2030
2030 DATA 1030,3010,3020
2040 DATA 1040,2050,3050
2050 DATA 1050,3030,3040
2060 DATA 1060,2070,2110
2070 DATA 1070,2080,2090
2080 DATA 1080,3060,3070
2090 DATA 1090,3080,2100
2100 DATA 1100,3090,3100
2110 DATA 1110,2120,3110
2120 DATA 1120,3120,3130
2900 REMark
2990 REMark Answers
3000 DATA 1510,0,0
3010 DATA 1520,0,0
3020 DATA 1530,0,0
3030 DATA 1540,0,0
3040 DATA 1550,0,0
3050 DATA 1560,0,0
3060 DATA 1570,0,0
3070 DATA 1580,0,0
3080 DATA 1590,0,0
3090 DATA 1600,0,0
3100 DATA 1610,0,0
3110 DATA 1640,0,0
3120 DATA 1620,0,0
3130 DATA 1630,0,0
```

answer is yes, and 'c' is the DATA line number should the response be no. If the variable 'b' READs the value zero, this indicates that an answer has to be found, and the loop is terminated. The subroutine at 5040 ensures that the question is neatly printed on the screen, while lines 5000 to 5120 print the answer, and ask you if you want to run through another set of questions.

Ambiguity

Like most 'experts', while I believe my questions to you (as placed in this computer program) are clear and unequivocal, you may well find them confusing. This expert has no option for the user to say "I don't understand that, please re-phrase the question" (also the case for most expert systems). Even the first question "Does it travel on land?" may provide uncertainty. A hovercraft certainly can travel on land, but I, as the 'transport expert', decided that hovercraft are used principally for sea transport, therefore would expect a 'no' answer to the 'land' question if the user were thinking of the features of a hovercraft.

But like this system or not, it is the way many human experts work. Not only that, the program I have given you provides an expert system 'shell', to which other knowledge bases can be attached. Fig. 3 contains an example. This is an alternative knowledge base which allows you, using the same program, to determine the name of one of the body's axial or appendicular bones. Lines 800 to 3130 replace the transport knowledge base in Fig. 1. Here, I assume, there are less experts among our readers, and you should, as non-experts get some benefit on terminology on human bones!

Once you feel you understand the logic of this expert system 'shell', there is no reason why you cannot build your own expert application. The first step is to draw up a decision tree, as shown in Fig. 2, then translate that to the data required for the knowledge base.

The more cautious among us would say that this approach to making decisions is for the foolhardy; it is a little rash to eliminate a whole range of possibilities on the outcome of just one question, which might be misunderstood. It might be more sensible to ask a full set of questions, then judge all the information at one time. This is the parallel approach to decision making, and this approach is the basis for decisions made by the program listed in Fig. 4.

There are several passages in the program in Fig. 4 which are identical to that in Fig. 1, so start with the serial decision-making program in your computer, and delete or overwrite lines 220 to 800, and the knowledge base 2000 to 2120. There is slight modification required to the questions (lines 1000 to 1090) and to line 5020. The commands in italics in lines 300 and 310 are, again, optional on the QL, and the string joining '&' operator in these lines should be replaced by '+' if you are using a Spectrum.

DATA line 800 now contains not only the title of our expert system, but two numbers, the number of questions and answers in our system. These two numbers set up a string array (s$) in line 240. This array contains zeros and ones, corresponding to expected no and yes responses to each question for all 13 modes of transport in our knowledge base.

All 10 questions are asked; in response to the user's reply, a variable (r$) is filled with zeros and ones (lines 300 and 310), to get a string the same length as the strings in the knowledge base, once all questions are answered. This reply string is then compared with the strings in the knowledge base.

Perfect Match

The first check (lines 330 to 380) is to determine whether a perfect match can be found, ie. a string of yes and no's matches exactly the expert's opinion of the expected string of yes and no's to each of the 13 outcomes. If a match is found, then the expert's answer is given.

If no match is found, the computer first tells you this, then goes through a routine (lines 420 to 500) to determine which set of expected yes/no replies in its knowledge base most closely matches the replies given by the user. To do this, it goes through each possible outcome, checking the expected response, with the one given, and awards a score, depending on the match. So, 1 is added to the score if a 'yes' answer was given, when one is appropriate for that particular item and

question (line 450), and 1 is subtracted for a 'yes' reply when 'no' was expected. In this way the item with the maximum 'score' is identified (line 480).

Logic might suggest that you should also award a point for two no's which match, and subtract one from the score when no is given to a question in which yes is expected from a certain outcome. Well, the scoring system used does work remarkedly well, without these 'logical' additions. In fact, as there are far less 'yes' answers than 'no' in every string in the knowledge base, matching 'yes' answers are more valid than matching "no's".

Another argument against this scoring system is that, for any outcome, certain questions are more important than others in differentiating items, so the amount added (or subtracted) should be weighted to reflect this. That argument would lead us into the realms of probability, which is something I shall save for later in the series!

One thing you'll note about parallel decision making is that a lot more questions will be asked, sometimes quite irrelevent when, in hindsight, you know the result. In the parallel decision program, you'll always be asked 10 questions, whereas, the serial decision program which solves the same problem asks between two and five questions before giving its result.

No doubt, a better transport 'expert' than myself could have selected a better, and smaller, set of questions to enable the 13 items to be differentiated in the parallel program. But as a general rule, parallel systems will always ask more questions than serial systems. But even if the data match isn't perfect, the parallel system is more likely to give you the right answer; one incorrect response with the serial system, and the expert's reply is certain not to be the one you want.

These are expert system decision making processes in their simplest form. Many expert systems allow more than yes/no answers; decision making can be mixtures of serial and parallel, depending on the problem. And these examples assume no uncertainty in your reply (or in the knowledge base).

These expert systems are also rather limited in their knowledge; there are many more than 13 different modes of transport. This 'expert' will have to learn a lot more about transport before it really can begin to describe itself as an expert. And how expert systems learn is the subject of the next part in this series.

Figure 4: QL/Spectrum Listing for Parallel Decision Demonstration

```
10 REMark Simple Expert System
20 REMark Using parallel decision-making
30 REMark
40 REMark David Nowotnik
50 REMark January, 1987
60 REMark
100 REMark Initialise & Title
110 LET x=2000
120 RESTORE : READ t$
130 CLS: PRINT t$
140 PRINT: PRINT
150 PRINT "I will ask you a series of"
160 PRINT "questions. Answer with y"
170 PRINT "for yes or n for no for"
180 PRINT "each one until I provide"
190 PRINT "you with the solution."
200 PRINT : PRINT
210 REMark
220 REMark Initialise
230 READ answ,quest
240 DIM s$(answ,quest)
250 RESTORE 2000
260 FOR j=1 TO answ: READ s$(j): NEXT j
270 LET r$="": RESTORE 1000
280 FOR j=1 TO quest
285 GO SUB 5140
290 INPUT z$: LET ch=0: IF z$="" THEN GO TO 290
300 IF z$(1)="y" OR z$(1)="Y" THEN LET r$=r$&"1":
LET ch=1: PRINT " yes"
310 IF z$(1)="n" OR z$(1)="N" THEN LET r$=r$&"0":
LET ch=1: PRINT " no"
315 IF ch=0 THEN GO TO 290
320 NEXT j
330 REMark Perfect Match?
340 DIM q(answ)
350 FOR j=1 TO answ
360 LET q(j)=0
370 IF r$=s$(j) THEN LET larg=j: GO TO 5000
380 NEXT j
390 PRINT: PRINT " I did not find a perfect match"
400 PRINT "with my data; I am searching"
410 PRINT "for the best fit."
420 LET max=0: LET larg=0
430 FOR i=1 TO answ
440 FOR j=1 TO quest
450   IF r$(j)="1" AND s$(i,j)="1" THEN q(i)=q(i)+1
460   IF r$(j)="1" AND s$(i,j)="0" THEN q(i)=q(i)-1
470 NEXT j
480 IF q(i)>max THEN LET max=q(i): LET larg=i
490 NEXT i
500 IF larg<1 THEN PRINT "No match found!": GO TO
5070
510 GO TO 5000
800 DATA "DETERMINE THE TYPE OF TRANSPORT",13,10
810 REMark
900 REMark Questions
1000 DATA "Does it travel on land"
1010 DATA "Does it travel on rails"
1020 DATA "Does it have two wheels"
1030 DATA "Is it powered by an engine"
1040 DATA "Can it carry more than 6 passengers"
1050 DATA "Does it travel more than 20 feet above
the surface"
1060 DATA "If it travels on water, does it only
float on the surface"
1070 DATA "Does it have fixed wings"
1080 DATA "Is it wind powered"
1090 DATA "Can it travel below the surface of
water"
1500 REMark Answers
1510 DATA "train"
1520 DATA "motor bike"
1530 DATA "bicycle"
1540 DATA "coach"
1550 DATA "car"
1560 DATA "aeroplane"
1570 DATA "helicopter"
1580 DATA "glider"
1590 DATA "balloon"
1600 DATA "yacht"
1610 DATA "ship"
1620 DATA "submarine"
1630 DATA "hovercraft"
1890 REMark
1900 REMark Knowledge Base
2000 DATA "1101100000"
2010 DATA "1011000000"
2020 DATA "1010000000"
2030 DATA "1001100000"
2040 DATA "1001000000"
2050 DATA "0001110100"
2060 DATA "0001110000"
2070 DATA "0010000110"
2080 DATA "0000010010"
2090 DATA "0000101010"
2100 DATA "0001101000"
2110 DATA "0001101001"
2120 DATA "1001100000"
2900 REMark
5000 REMark Provide the answer
5010 PRINT
5020 RESTORE (larg*10+1500): READ e$
5030 PRINT "The answer is a";
5040 IF e$(1)="a" OR e$(1)="e" OR e$(1)="i" OR
e$(1)="o" OR e$(1)="u" THEN PRINT "n ";: GO TO 5060
5050 PRINT " ";
5060 PRINT e$
5070 PRINT: PRINT "Any more? (y/n) ";
5080 INPUT z$
5090 IF z$="" THEN GO TO 5080
5100 IF z$(1)="y" OR z$(1)="Y" THEN RUN
5110 IF z$(1)="n" OR z$(1)="N" THEN STOP
5120 GO TO 5070
5130 REMark
5140 REMark Print question string
5150 READ e$
5160 LET i=0: LET LE=LEN(e$): LET LI=0
5170 LET x$=""
5180 LET i=i+1: IF i>LE THEN GO TO 5210
5190 IF e$(i)=" " THEN GO TO 5210
5200 LET x$=x$&e$(i): GO TO 5180
5210 IF i>LE THEN LET x$=x$&"?"
5220 IF LI+LEN (x$)>31 THEN PRINT: LET LI=0
5230 PRINT x$;: LET LI=LI+LEN (x$)
5240 IF LI<31 THEN PRINT " ";: LET LI=LI+1
5250 IF i<LE THEN GO TO 5170
5260 RETurn
```

Exploring PRINTER GRAPHICS

Alan Davis explores the possibilities for printing graphics.

In a recent issue of ZXC, I wrote a short piece about the Kempston "E" interface and Amstrad DMP 2000 printer, specifically to point out how painless I'd found it to set up and use this combination. Of course that isn't the end of the tale, because obviously you can do a great deal more with such a system that merely LPRINT, LLIST, and COPY. What I'd like to do in this article is to explore some of the graphics possibilities of the system, and since the Amstrad is Epson compatible, the routines should work perfectly well with any Epson-type printer which is capable of small (1/216") line feeds.

Regular readers may be surprised to discover that the origin of what follows has nothing to do with adventure games at all (Gasp! Shock!), but the fact is that I also use the Spectrum for serious scientific work — and one of my requirements for this is to be able to get hard copy of graphs of fairly complex mathematical functions. Now if the graph you want to plot is fairly simple, there's no problem. You can just write a program to draw it on screen, and then COPY the result. But the screen resolution of 256x176 pixels is too coarse for the applications I have in mind, and so some way of getting higher resolution had to be found.

The approach I adopted was fairly obvious: instead of drawing the "y" axis of the graph vertically on the screen, I turned it through 90 degrees and drew it horizontally, so that the "x" axis becomes vertical (with x increasing downwards). The graph is then plotted eight points at a time (i.e. eight successive x values) to line 21 on the screen, the whole screen being scrolled upwards by one line between the plotting of one batch of eight points and the next. In this way you can get any resolution you like along the x-axis, if necessary plotting and scrolling many screenfuls of graphics before the whole process is finished. Naturally, though, you'll never see the entire graph on screen at once, because the early parts will have scrolled off the top of the screen by the time the thing has finished.

What we need then, is a method of transferring just a single line of graphics data (line 21 in this case) from the screen to the printer, so that we can dump it out continuously in between point plotting and screen scrolling. But sadly, there's no single printer command for doing this. The Kempston interface will give you a full screen copy, but not just one line, and so a bit of programming is necessary to achieve this.

Bit images

The key to the problem lies in the "Bit Image Mode" printer commands. There's a fair choice of these on the Amstrad machine, and in the end I settled for "double density" mode. This gives a printed dot spacing which is half that obtained with "single density" mode — and therefore a printed image of higher quality and crispness. The relevant command is ESC L + n1 + n2. Translated into English this means that we need to send to the printer the codes CHR$ 27 (ESC), followed by CHR$ 76 ("L"), followed by two mysterious codes n1 and n2. These last two codes tell the printer how many bytes of graphic data are to follow, and together they form a two-byte number where n1 represents the low byte, and n2 the high. So if you want to print, say, 258 bytes of data in

Listing 1

```
?ass 1 errors: 00

                10 *D+
        20 *C-
        30 ;Routine for copying screen line 21 to printer
        40 ;Any given pixel has its screen coordinates x and y
        50 ;held in the C and B registers respectively
        60
65368    70         ORG  65368 ;Start of UDG area
65368    80         LD   A,3
65370    90         CALL #1601 ;Select the printer for output
65373   100         XOR  A
65374   110         LD   D,A ;Clear D register
65375   120         LD   C,A ;Clear C register (x coord =0)
65376   130 LOOP1   LD   B,8 ;Initialise B register (y coord =8)
65378   140 LOOP2   DEC  B ;Next pixel row down
65379   150         PUSH BC
65380   160         PUSH DE ;Save registers
65381   170         CALL POINT ;Same as POINT (c,b)
65384   180         CALL FP_A ;Result off stack into A register
65387   190         POP  DE
65388   200         POP  BC ;Recover registers
65389   210         CP   1 ;Is this pixel (c,b) set?
65391   220         CALL Z,SETBIT ;If so, then set bit 7 of D
65394   230         RLC  D ;Rotate D register by one bit
65396   240         INC  B
65397   250         DJNZ LOOP2 ;Last pixel? If not, do another
        260
        270 ;At this point we have a byte ready to print in D
65399   280         LD   A,D
65400   290         RST  16 ;So print it once...
65401   300         LD   A,D
65402   310         RST  16 ;...and once again
65403   320         LD   D,0 ;Clear D reg. ready for next scan
65405   330         INC  C ;Next pixel column to the right
65406   340         LD   A,C
65407   350         CP   0 ;Are we off the right edge of screen?
65409   360         JR   NZ,LOOP1 ;If not, back for another byte
65411   370         RET  ;Back to BASIC
        380
65412   390 SETBIT  SET  7,D ;Set bit 7 of D register
65414   400         RET
 8910   410 POINT   EQU  #22CE ;(ROM)
11733   420 FP A    EQU  #2DD5 ;(ROM)

Pass 2 errors: 00

Table used:    73  from   249
```

Listing 2

```
   1 REM *** CONTINUOUS GRAPH PRINTING ROUTINE
   2 REM
   3 DEF FN f(x)=150*SIN (x/6)*SIN (x/60)*EXP (-x/200)
   4 GO SUB 1000
   7 REM
   8 REM *** Kempston "E" - Tokens off
   9 REM
  10 COPY : REM CHR$ 0
  20 LPRINT CHR$ 27;"A";CHR$ 0;: REM * Disable auto line feed
  27 REM
  28 REM ***Lines 30 - 160 draw the function
  29 REM
  30 LET oy=127: PLOT 0,7: DRAW 255,0: REM * Draw y-axis
  40 FOR i=0 TO 599 STEP 8: REM * 600 values of x
  50 FOR j=0 TO 7: REM * x values in bunches of 8
  80 LET y=127+INT (.5+FN f(i+j)): REM * (i+j)=x coordinate
  90 PLOT 127,7-j: PLOT y,7-j: REM * Extend axis:Plot next point
 100 LET len=oy-y: IF ABS len>1 THEN  DRAW len-(1 AND len>0)+(1
AND len<0),0: REM * Join the dots
 110 LET oy=y: REM * Store current value of y
 120 IF INT ((i+j)/10)=(i+j)/10 THEN  PLOT 126,7-j: DRAW 2,0: IF
 INT ((i+j)/50)=(i+j)/50 THEN  PLOT 123,7-j: DRAW 8,0: REM * Sca
le
 130 NEXT j
 140 GO SUB 200: REM * Line finished - print it out
 150 RANDOMIZE USR 3582: REM * Scroll screen up one line
 160 NEXT i
 170 STOP
 197 REM
 198 REM *** Print out screen line 21
 199 REM
 200 FOR h=0 TO 2: REM * 3 passes of printhead
 205 LPRINT CHR$ 27;"L";CHR$ 0;CHR$ 2;: REM * Bit image mode
 210 RANDOMIZE USR 65368: REM * Print the line
 215 LPRINT CHR$ 13;CHR$ 27;"J";CHR$ 1;: REM * Small line feed
 220 NEXT h
 230 LPRINT CHR$ 13;CHR$ 27;"J";CHR$ 21;: REM * Big line feed
 240 RETURN
 500 DATA 62,3,205,1,22,175,87,79,6,8
 510 DATA 5,197,213,205,206,34,205,213,45,209
 520 DATA 193,254,1,204,132,255,203,2,4,16
 530 DATA 235,122,215,122,215,22,0,12,121,254
 540 DATA 0,32,221,201,203,250,201
 997 REM
 998 REM *** Poke machine code
 999 REM
1000 RESTORE 500: FOR i=65368 TO 65414: READ x: POKE i,x: NEXT i
1010 RETURN
```

graphics mode, then n1 would need to take the value 2, and n2 and value 1 — because (2+1x256) = 258. Yes? Right!

Now, to copy a single line of the screen (eight pixels high and 256 pixels wide), we need to send 256 bytes of data to the printer — but since we'll be printing in double density mode this would produce a final image at only half size, which is not so good (dot spacing is halved in this mode, remember).

So, to get a full-size printed image, then, all we need to do is to send each byte twice — and so we need to tell the printer to expect not 256 bytes, but 512, to be printed in graphics mode. So the appropriate escape sequence for our purpose becomes:

CHR$ 27; "L"; CHR$ 0; CHR$ 2;

The next job is to scan horizontally through line 21 on the screen one pixel at a time, forming a single byte of data from each vertical column of eight pixels and then shoving it out to the printer. You can do it in BASIC — but frankly, life's too short! Assuming you'd like to see your hard copy this century rather than next, we'd better do it in machine code, and I've given in Listing 1 an assembler program which will solve our problem. I've annotated the listing in some detail so that you should be able to follow it through without too much difficulty. (As written, the code is organised at the start of the user-defined graphics area: 65368).

Strictly speaking, all we need to add to this is a short BASIC subroutine which puts the printer into the correct bit image mode, calls the machine code to transfer the 512 bytes of screen line data, and then tells the printer to do a carriage return and a line feed of 8/72" (equivalent to eight vertical dot spacings) at the end, thus:

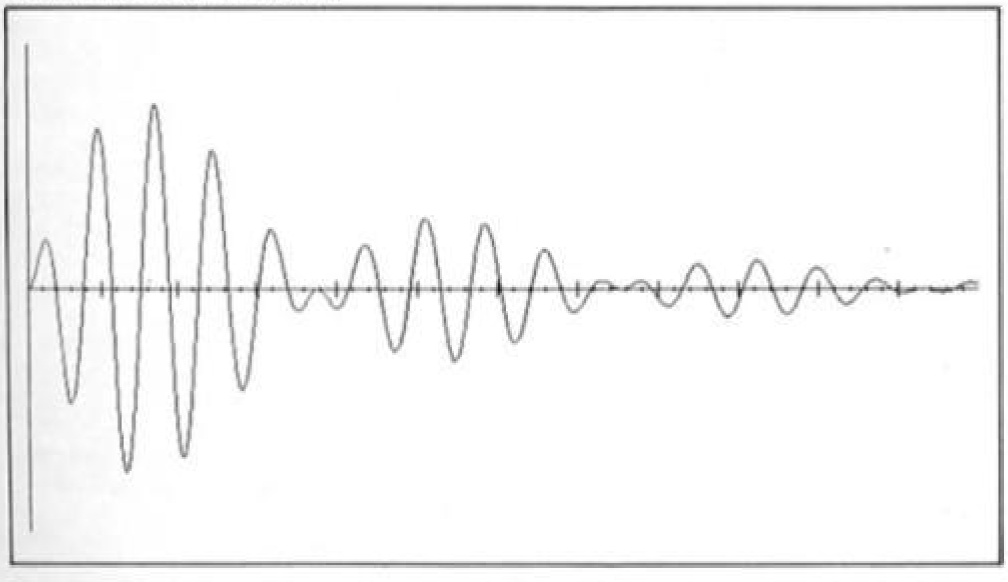
1) LPRINT CHR$ 27; "L"; CHR$ 0; CHR$ 2; (See above)
2) RANDOMISE USR 65368 (Send the data)
3) LPRINT CHR$ 13; CHR$ 27; "A"; CHR$ 8; (CR then 8/72" LF)

We could then call this subroutine after every eight plotted points, and away we go. Well, yes — but we can actually do very much better. You see, the final printed result from this sequence would turn out to be a rather "spidery" affair because we're using only single density dot printing in the "x" direction. Fortunately it's a simple matter to sort this out. All we have to do is to send the same data from the screen line to the printer three times in succession with a very small line feed of 1/216" (1/3 vertical dot spacing) after each transmission. At the end of this, we send a line feed of 21/216" (making a total of 24/216", or 8/72") to get the printer ready for the next line of data. What we achieve by this is effectively triple dot density in the x direction.

Graphs

Listing 2 gives an example application of this process, the important part being the subroutine at line 200 which employs all the little tricks we've just been discussing. The program as a whole will produce a graph of the function defined in line 3 (which is of no importance except as an illustration), and Figure 1 shows the final printed result. Incidentally, the machine code is so short that it's hardly worthwhile loading it in separately — in fact line 1000

Figure 1. Hi-res graph plotting



Listing 3

```
Pass 1 errors: 00

              10 *D+
       20 *C-
       30 ;Routine for printing a vertical strip of screen 8
       40 ;pixels wide, starting at the pixel whose x-coord
       50 ;is held in address XCOUNT
       60
65368  70            ORG  65368 ;Start of UDG area
65368  80            LD   A,3
65370  90            CALL #1601 ;Select printer
65373  100           XOR  A
65374  110           LD   B,A
65375  120           LD   D,A ;Zero the B and D registers
65376  130 LOOP1     LD   A,(XCOUNT)
65379  140           LD   C,A ;Starting position in C register
65380  150 LOOP2     PUSH BC
65381  160           PUSH DE ;Save registers
65382  170           CALL POINT ;Equivalent to POINT (c,b)
65385  180           CALL FP_A ;Result off stack into A register
65388  190           POP  DE
65389  200           POP  BC ;Recover registers
65390  210           CP   1 ;Is this screen pixel set?
65392  220           CALL Z,SETBIT ; If so then set bit 7 of D
65395  230           RLC  D ;Rotate D register by one bit
65397  240           INC  C; Move to next pixel on the right
65398  250           LD   A,(XCOUNT)
65401  260           ADD  A,8
65403  270           CP   C ;Have we done 8 pixels yet?
65404  280           JP   NZ,LOOP2 ;If not then process this one
       290
       300 ;D register now holds a byte ready to print
65407  310           LD   A,D
65408  320           RST  16 ;So print it once...
65409  330           LD   A,D
65410  340           RST  16 ;...and once again
65411  350           LD   D,0 ;Clear D register
65413  360           INC  B ;Move up the screen by one pixel
65414  370           LD   A,B
65415  380           CP   176 ;Are we past the top of the screen?
65417  390           JP   NZ,LOOP1 ;If not then get another byte
65420  400           RET  ;Back to BASIC
       410 SETBIT
65421  420           SET  7,D ;Set bit 7 of D register
65423  430           RET
65424  440 XCOUNT    DEFB 0 ;x coord of left edge of strip
 8910  450 POINT     EQU  #22CE ;(ROM)
11733  460 FP_A      EQU  #2DD5 ;(ROM)

Pass 2 errors: 00

Table used:    86  from   256
```

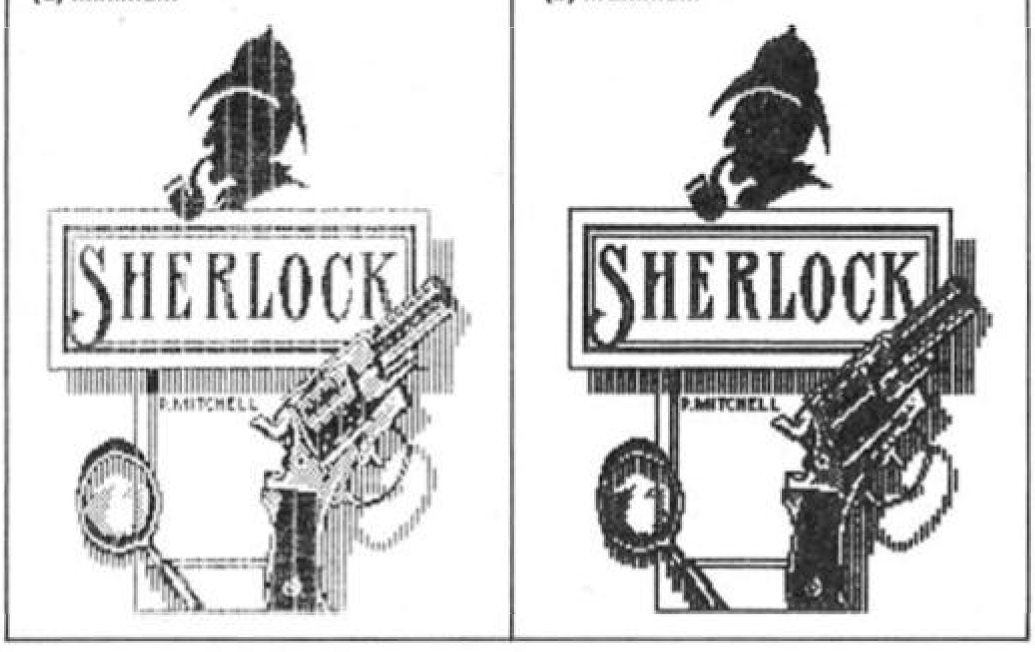
just pokes it in from BASIC when the program runs. Just a couple of further points about the listing: (a) You need to instruct your interface to turn off the tokens before proceeding, and line 10 does this for the Kempston "E" interface. (b) Line 20 is needed only if your printer is normally set to give automatic line feeds with a carriage return.

You may wonder why I've left so much of the printer handling to BASIC, and there are two reasons for this. First, given the machine code routine to transfer the data, it's the actual physical process of printing which takes the bulk of the time, so that there's very little to be gained by writing the remaining bits in machine code. Second, with the subroutine as it stands you can fiddle about with it easily. You might, for example, decide to go for only two passes of the print-head rather than three, in order to speed up the printing — and if you do you'll need to change the counter in line 200 (FOR h = 0 TO 1), and to change the line feed in line 230 to 22/216".

Figure 2. Variable density screen copy

(a) Minimum

(b) Maximum



Stripper

So much, then, for high resolution graph plotting. Is there any other use for this idea of printing strips of screen in double density dots? There certainly is. How would you like a variable density screen copy routine so that you can compensate for the wear on your printer ribbon? You would? Then read on!

The Kempston "E" interface allows you to get two kinds of screen copy — an enlarged one which is fine for most purposes, and one of "normal" size which is OK but could be better. This latter is a single density screen dump (which looks rather ghostly when your ribbon is worn) with a quite badly distorted aspect ratio to boot. If we take command of the printer ourselves, and scan the screen in strips (treating each strip much as we did in the graph plotting example), we can improve matters considerably.

There's some advantage to be gained by scanning the screen in vertical strips rather than horizontal ones in order to get a more acceptable aspect ratio, and as before, it's only necessary to handle the transfer of data from screen to printer in machine code. Listing 3 gives the assembler program we need. Though similar to Listing 1, it differs in some important ways namely (a) It transfers data from a vertical strip eight pixels wide and 176 pixels high; and (b) We need to tell it which strip to transfer by poking a suitable x co-ordinate into 65424 (XCOUNT). Again, the code itself is so short that it's readily POKEd in from within a BASIC program, and I've incorporated it in this way into Listing 4.

Listing 4 is a self-contained

screen copy routine which gives you the choice of three printing densities (effectively you're choosing the number of print-head passes between the line feeds). Just type it in and save to auto-run from line 1. In use the program assumes that the screen you wish to copy is to be loaded from microdrive, but if you're working with tape just change line 60 accordingly. Whichever density you choose, you'll get a result of higher quality than the normal Kempston screen copy would give — and maximum density will give excellent results with even the most battered and weary of printer ribbons. Figure 2 demonstrates the range of densities available and, even with the unavoidable distortions of reproduction, may give you some idea of the possibilities.

In this article I've only scratched the surface of what can be achieved, but I hope it's enough to give you some ideas — and in any case I find the routines I've given here pretty useful, myself. With only a few modifications, you should be able to produce "enlarged" versions using quadruple density mode, if you require them. And if nothing else, the maximum density copy routine might help you to squeeze just a little bit more out of your tired and worn-out printer ribbons.

Listing 4

```
   1 REM ***KEMPSTON "E"/DMP 2000 VARIABLE DENSITY SCREEN COPY
   2 REM
   3 REM
  10 CLS : PRINT AT 10,5;"SWITCH ON THE PRINTER": GO SUB 1000
  15 PRINT AT 21,3;"Press any key to continue": PAUSE 0
  17 REM
  18 REM *** Kempston "E" instruction to switch off tokens
  19 REM
  20 COPY : REM CHR$ 0
  30 LPRINT CHR$ 27;"A";CHR$ 0;: REM * Disable auto-line feed
  40 CLS : INPUT "Name of screen to be loaded?"' LINE z$
  50 IF z$="" THEN  GO TO 40
  60 LOAD *"m";1;z$SCREEN$
  70 PRINT #1;AT 0,0;"SELECT DENSITY 1/2/3 (3 maximum)""PRESS 4
to load another screen  "
  80 PAUSE 0: LET i$=INKEY$: IF i$<"1" OR i$>"4" THEN  GO TO 80
  85 BEEP .1,30
  90 IF i$="4" THEN  GO TO 40
  95 PRINT #1;AT 0,0,,,,AT 1,11; FLASH 1;"PRINTING"
 100 LET density=(1 AND i$="2")+(2 AND i$="3")
 110 FOR i=0 TO 255 STEP 8: FOR j=0 TO density
 120 POKE 65424,i: REM * Position of left hand side of strip
 130 LPRINT CHR$ 27;"L";CHR$ 96;CHR$ 1;: REM * Double density
 140 RANDOMIZE USR 65368: REM * Print a strip
 150 LPRINT CHR$ 13;CHR$ 27;"J";CHR$ 1;: REM * Small line feed
 160 NEXT j
 170 LPRINT CHR$ 13;CHR$ 27;"J";CHR$ (23-density);: REM * Big LF
 180 NEXT i
 190 GO TO 70
 997 REM
 998 REM *** Machine code for printing vertical strip of screen
 999 REM
1000 FOR i=65368 TO 65424: READ x: POKE i,x: NEXT i: RETURN
1010 DATA 62,3,205,1,22,175,71,87,58,144
1011 DATA 255,79,197,213,205,206,34,205,213,45
1012 DATA 209,193,254,1,204,141,255,203,2,12
1013 DATA 58,144,255,198,8,185,194,100,255,122
1014 DATA 215,122,215,22,0,4,120,254,176,194
1015 DATA 96,255,201,203,250,201,0
```

GREAT GRAPHICS

Here it is, the ZX guide to the best in graphics equipment for the Spectrum.

Art Studio £14.95

Unless you're going to stick to BASIC (not a good idea) the first thing you'll need when you start experimenting with graphics is a good bit of software, and, with one possible exception, Rainbird's Art Studio stands head and shoulders above everything else. Despite the wretched Lenslok anti-piracy device, Art Studio is a fast and flexible icon-driven package that will help you squeeze the best out of the Spectrum's graphics capabilities. It's also available in two versions for both 48K and 128K machines (although the 128 version costs £24.95).

The Artist II £14.95

Softek's original Artist program was good, but a bit fddly to use, but the enhanced Artist II has adopted the same pull-down menu system as Art Studio and there's now not a lot to choose between the two programs when it comes to creating screen graphics. However The Artist allows you to take text files from Softek's Writer wordprocessor and add graphics to them in order to create a passable imitation of a magazine page — desktop publishing on the Spectrum!

Cheetah Mach 1+ £14.95

Once you've picked your software you then have to decide how you're going to 'draw' your pictures on the screen. All graphics packages allow you to use the keyboard for positioning a drawing cursor, but this can get a bit awkward unless you've got the fingers of a concert pianist so a joystick can come in handy. People's choice of joystick is fairly personal, but we've found this one to be good for both delicate drawing movements and all-out alien zapping, and it has the additional advantage of having an adaptor that allows you to use it with both Kempston interfaces and the +2's built in joystick socket. Of course if you don't have a Plus Two you'll need...

Kempston Joystick Interface £6.95

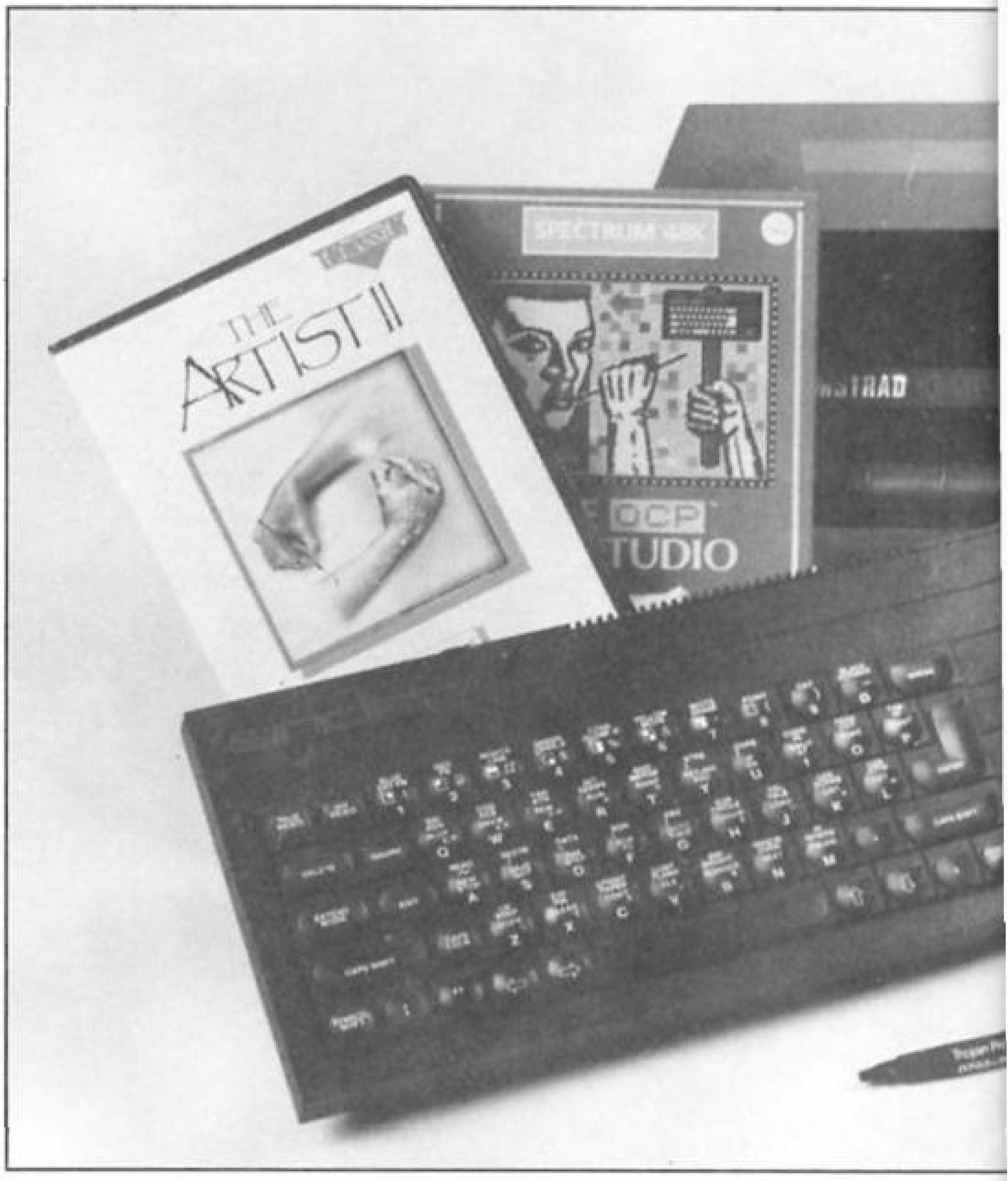
Almost as old as the Spectrum itself is the Kempston Joystick interface, more or less the *de facto* standard for the Spectrum. This latest model from Kempston is small, neat, and very inexpensive.

AMX Mouse £69.95

Everyone's got a joystick (haven't they?) but the more ambitious among you might want to splash out on a mouse. Most 'real' business computers have used mice for years, and there are now a couple available for the Spectrum (with another on the way from Saga quite soon). The AMX Mouse, from Advanced Memory Systems, appeared shortly before Rainbird released Art Studio and, being incredibly sensible, Rainbird made Art Studio compatible with the mouse (though you don't have to buy Art Studio as well since the mouse comes complete with its own bundled graphics software, AMX Art). The mouse interface plugs into the Spectrum just like a joystick interface, into which the mouse is plugged. A good value-for-money feature is that the mouse interface also includes a printer interface which could save you some £30 if you haven't already got one.

Trojan Light Pen £14.95

Not as flexible as a mouse, but less expensive, light pens have the virtue of allowing you to 'draw' directly onto the surface of your monitor or TV screen. This can be useful if you're trying to draw curves, wiggly lines etc. which can't easily be drawn using keyboard or joystick control, as the pen allows you to draw freehand. Trojan's pen



comes with its own graphics software, though this isn't as sophisticated as it could be.

Kempston Mouse £49.95

The price of this is about to come down to around £49.95 according to Kempston, making it excellent value. At the moment the mouse comes bundled with a copy of the Art Studio, making it about as powerful a graphics tool as you're likely to get on the Spectrum (though when the price comes down Kempston are intending to replace Art Studio with their own graphics software).

The mouse itself is a sleek white unit with two buttons set into its 'head'. It's very solidly constructed, and wouldn't look out of place beside a computer ten times as powerful as a Spectrum. Like the AMX mouse this also plugs into the Spectrum via its own interface which, not surprisingly looks just like the rest of the Kempston range of interfaces.

Kempston 'E' Interface £34.95

Where there's an interface there's bound to be a Kempston, and the Kempston 'E' comes highly recommended by ZX regular Alan Davis. All the software you need to run it with is held onboard, it's versatile enough to meet just about all your needs and, most importantly, it's simple enough to use that you don't have to be a genius in order to wade through endless lists of control codes — unlike a lot of printer interfaces this one is actually 'user-friendly'!

Euroelectronics ZXLprint III £34.95

We had to give this one a mention since we've had one in our offices for a couple of years and never had a moment's trouble with it (this article was dumped onto our printer with it). It's compatible with both Centronics and RS232 printers (you just have to buy the relevant cable), and like the Kempston 'E' it has its own onboard software.

What about a printer?

Oh dear. This is where it could start to get a bit complicated. Once upon a time you had a choice between a big expensive 'real' printer or a number of small cheap 'dedicated' printers which were specifically intended for use with the Spectrum (Sinclair's own ZX printer, the Alphacom 32 and Seikosha GP50-S). Unfortunately none of these are still in production, 'though you might be able to find one or two still floating around in shops.

So, you no longer have any choice — if you want to print out any of your graphics masterpieces you're going to have to fork out somewhere in the region of £200 for a full-size printer and interface.

A printer is a fairly serious piece of hardware and, in general, the price reflects the quality of the machine, although few Spectrum owners are likely to need to typewriter quality print offered by the top-range machines costing £400 upwards. If you're only interested in printing screen dumps you probably don't need near letter quality printing, but it's not going to cost you much extra and is bound to come in useful one day — especially if you ever want to sell your printer as second hand, so bear that in mind. However, there are a few printers that offer high quality at relatively low prices, our particular favourites being:



Amstrad DMP2000 £159

About as cheap as you're likely to get, and offering surprisingly good quality. Like most Amstrad products they've kept it simple and cheap, so this looks a bit like a baby compared to some of the heftier numbers around but it will do screen dumps and near letter quality text printing — and for under £200 that's good going.

Compliment LQ £199

Part of Saga's Compliment word processor package this is an excellent machine that's as good as some much more highly priced printers. Obviously geared towards wordprocessing rather than graphics, but still capable of good quality screen dumps.

Citizen 120D £235

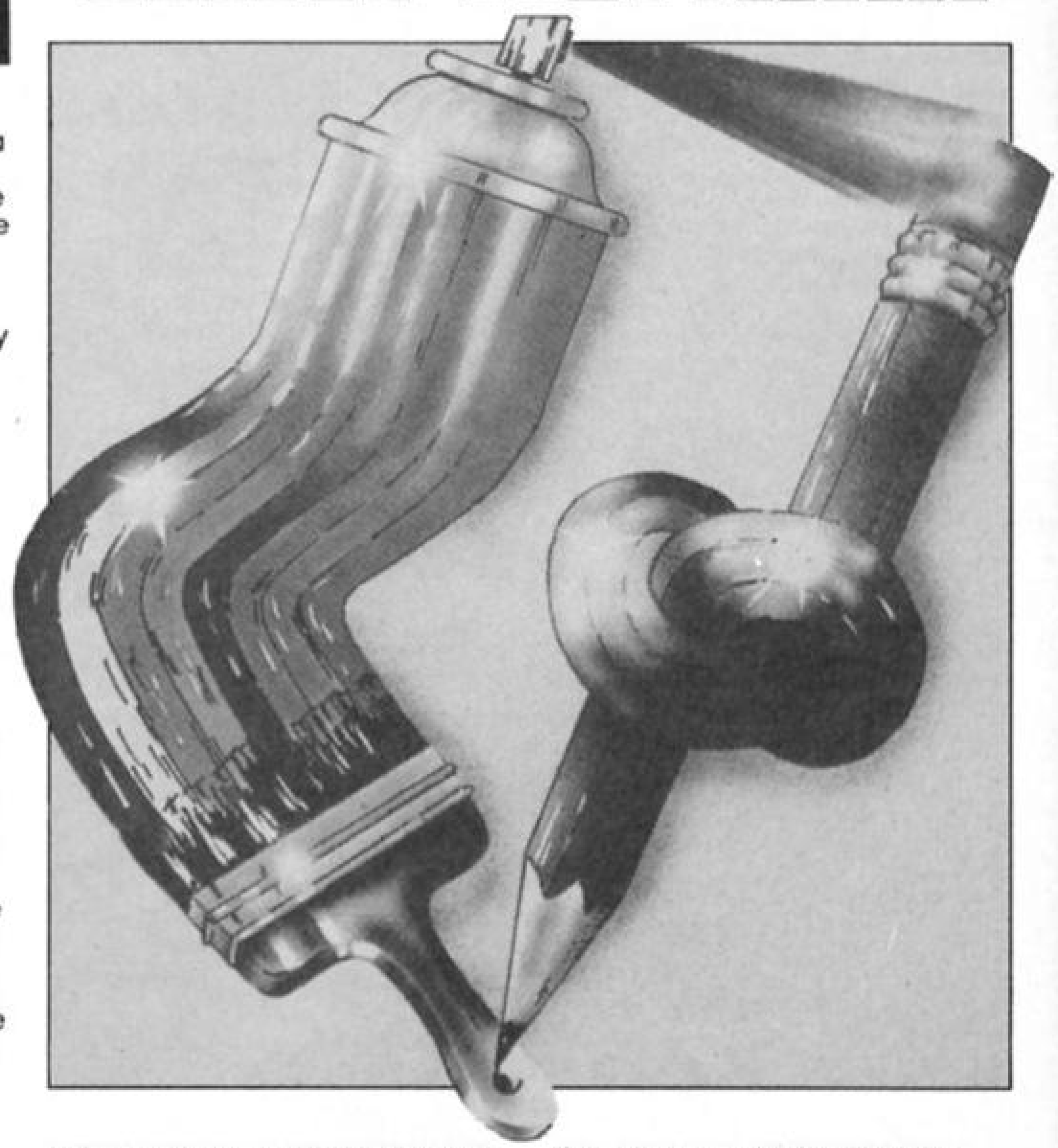
Starting to get a little pricey, but this is about as good as most home computer owners will need. There is a very good letter quality mode for printing text, and it's crammed with international character sets and self-diagnostic tests, and takes single sheets of paper or perforated printout paper. Some of these features are luxuries, rather than necessities, but if you can afford it you're not likely to have to buy another printer very soon.

GREAT GRAPHICS

ICON GRAPHIX 128

Hot on the heels of The Advanced Art Studio comes Audiogenic's graphics package for the 128 and Plus 2.

Icon Graphix 128
Audiogenic
£12.95



Audiogenic's Icon Graphix package was launched about a year ago, at the same time as the Spectrum 128 appeared. The first version to appear was for the 48K machines but, not surprisingly, Audiogenic also announced that an enhanced version for the 128 was on its way and, now that Amstrad's own version of the 128 is well and truly here, they've finally gotten around to releasing Icon Graphix 128.

New features

Unlike the enhanced version of Art Studio, which uses the 128's additional memory mainly for fast data storage/retrieval and which has relatively few new facilities for actually creating graphics, the new Icon Graphix package features a number of options which were absent from the original. Mind you, the original Icon Graphix lacked certain features that could be found in both Art Studio and The Artist (such as text handling and 'cut and paste' commands) so the upgraded version has been brought into line with what we've all come to expect as standard from graphics packages these days.

The layout of the screen display remains unchanged, with the central drawing area surrounded by a border made up of the various FILL patterns and, along the bottom of the screen, icon symbols for the graphics commands. This icon menu now has an added feature though — a PAGE icon, which when selected calls up pages 2 and 3 of the menu containing the new commands.

The FILL, PEN, and shape commands (for drawing boxes, lines, circles and ovals) were all standard on the 48K version but the new commands include some for adding text to your pictures (and there is an impressive variety of fonts included on side two of the tape, although only one font at a time can be held in memory).

The other new commands are mainly for 'cut 'n paste', allowing you to define small sections of the screen which can be 'lifted up', moved around, rotated, reflected and otherwise abused. As I mentioned these can be found on ordinary 48K packages, so while their inclusion now is a good idea it's not exactly a giant leap forward.

Trace

One new feature which is quite original is the TRACE command. Once you've defined a window (or a 'cutting' as the manual calls it) you can use this command on whatever is drawn within it. What it does is to invert the paper and ink within the cutting and to then surround any solid areas with either a black or white outline depending on the nature of the drawing. This might sound a little pointless but it actually creates a kind of 'polarised' effect that makes even simple drawings look more interesting, and it can be used repeatedly to build up some unusual shading effects which might otherwise take forever to produce if they had to be drawn conventionally.

This TRACE command is the highlight of the package, and, along with the icon menus (which I found simpler to use than on Art Studio or The Artist) is one of the areas where it scores over the competition. However



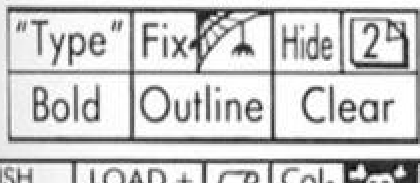
there are other features which could be improved upon given the 128's potential.

Although the program allows you to control the cursor from keyboard, joystick or (Kempston) mouse, the movement of the cursor is fairly slow and it can take a couple of seconds to move the cursor from one side of the screen to another (good if you're doing detailed work using the MAGNIFY option, but irritating if you're trying to quickly move from one icon to another).

The FILL and shape drawing commands are also slow, and if you're defining a cutting with the cut and paste commands that is more than about 1/4 of the size of the screen you can almost hear the poor machine straining to keep up as it tries to constantly redraw the window. Finally, the SAVE/LOAD facilities are a bit of a mess. You can save or load a normal SCREEN$, a 'document' or a cutting, though the manual doesn't really explain how a document differs from a SCREEN$. The loading option is a bit finicky about accepting names and you can't just load the first SCREEN$ that you come to on tape (ie using LOAD "" SCREEN$), as you've got to know the name of the screen to start off with. This is fine if you've saved a picture drawn from scratch on the package, but if you've got some pictures saved onto tape that are taken from anywhere else you've got to make sure you know their names before you can get at them. Icon Graphix 128 *is* an improvement upon the original. I found it easier and more enjoyable to work with than many other graphics packages I've had to review in the past, mainly because the icon menu is so self-explanatory and visible on screen all the time rather than just popping up and down before you've managed to figure out what you're doing.

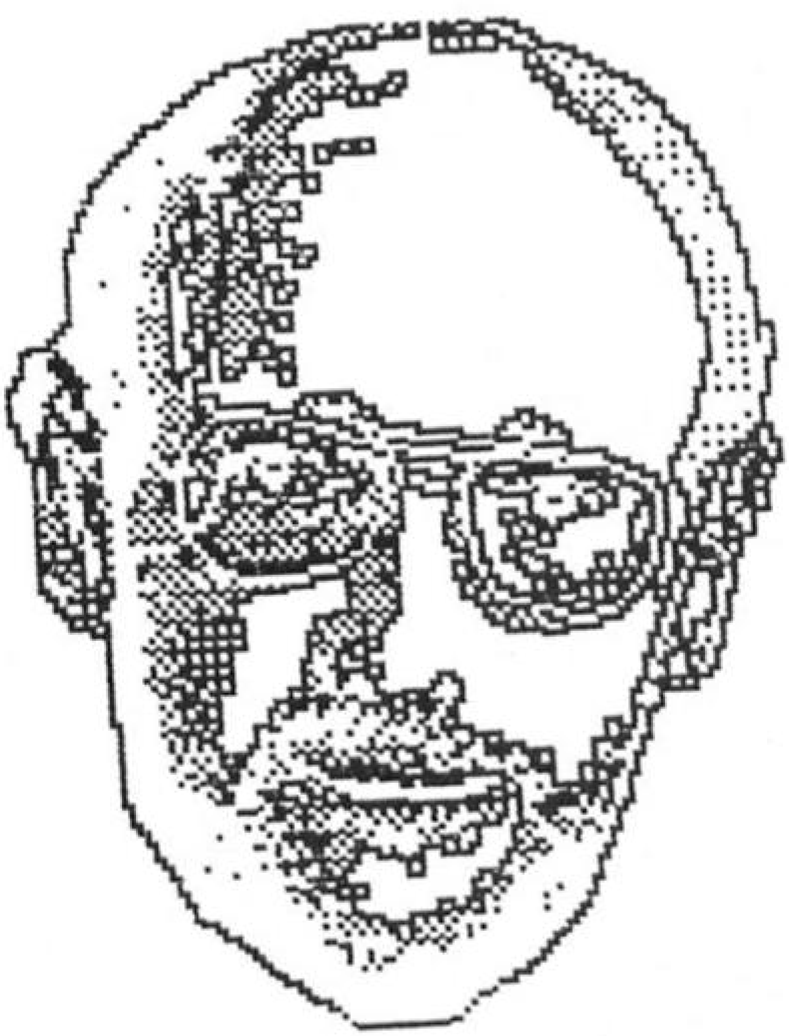
As I mentioned, the program could be faster than it is, but on the other hand it does cost just £12.95 compared to a fairly hefty £25 for the 128 version of Art Studio, so while it might not match Art Studio in sheer speed and number of options it still works out at quite good value.







Above and below: Sir Clive before and after being 'traced'.





DISCIPLE

John Wase plugs in a new multi-purpose interface.

Disciple
Rockford Products
£73.70 (+ VAT)

For several weeks I had been anticipating its arrival. At last a cryptic message came — "a disciple will shortly arrive at your house". It was on its way.

My delay in receiving this interface was due to a technical problem: the NEC ROM version of the Spectrum gave trouble with their first issue, and the ROM in the Disciple had to be rewritten. Advantage was therefore taken of this to incorporate several improvements into the system, and it is this version, version 2, which I tested. Disciple comes complete with instruction manual, guarantee and a cassette. As the picture shows, the interface itself is a long, flat plastic box on which your Spectrum sits, a bit Interface One-ish in shape, with an edge connector on top to connect the Spectrum and the customary through-port on the back. To the right of this are the joystick port and network connector, with the printer interface on the end of the box. To the left are a standard BEEB-type disc interface connector, a second joystick port and network connector, and on the left-hand end, the inhibit and snapshot buttons.

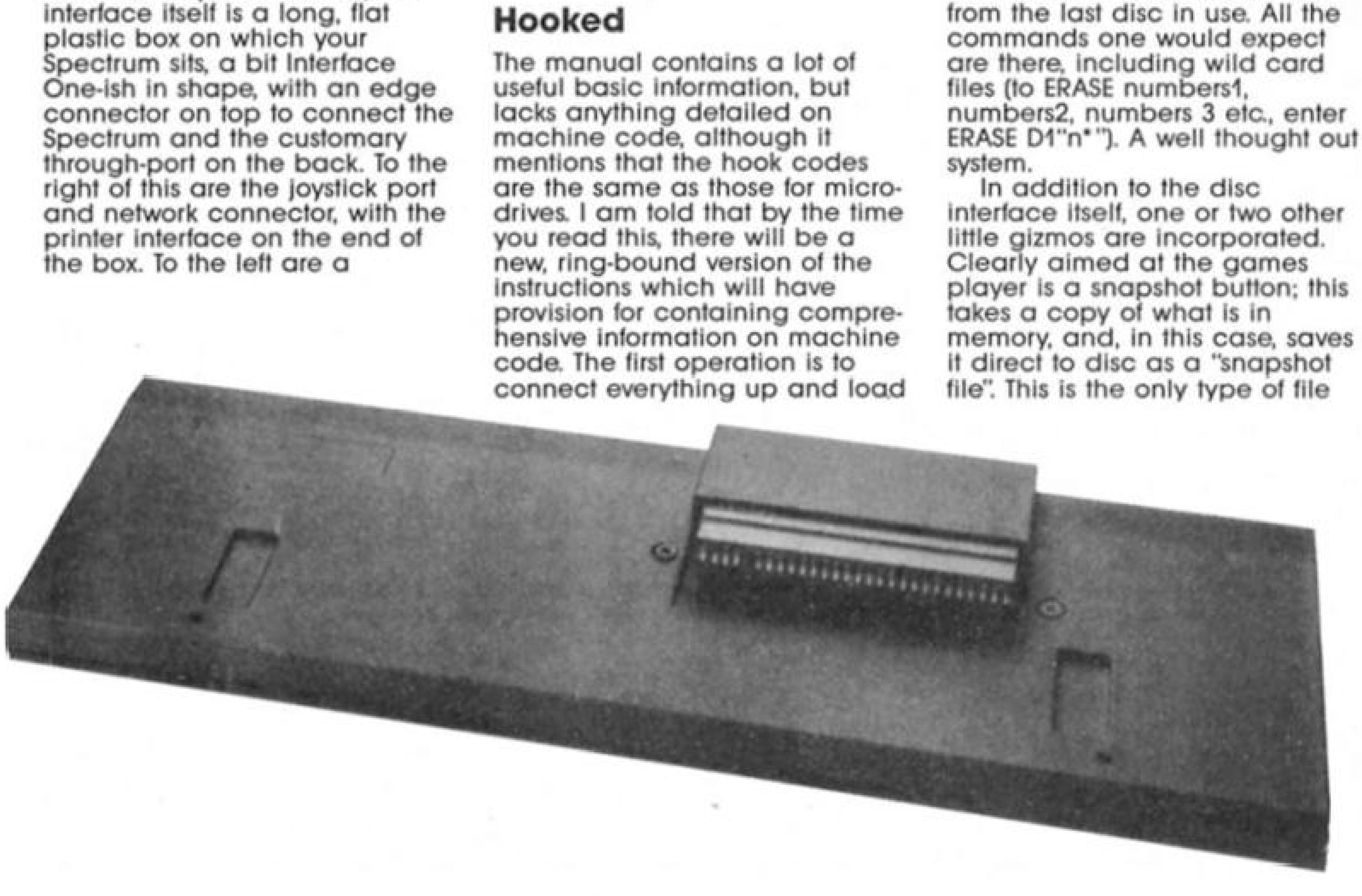
I approached the system with some trepidation, being prejudiced through long familiarity with the Opus system. So at first I didn't like the rather creaky plastic box; a 128K Spectrum+2 overhung this rather a lot, and I was glad I had left on the big feet which I need with the Discovery, for they just supported in the right places the overhang from the Disciple. A standard 48K Spectrum was less lucky and needed a cassette temporarily wedged in front of it. The first two Disciples had rough handling in the mail and the snapshot switch was damaged in each case; Rockfort tell me that more robust switches are now being used, and that the packaging is being modified. Whilst none of this improved my confidence, as I used it and gradually got used to it, I began to like it and appreciate its many virtues.

Hooked

The manual contains a lot of useful basic information, but lacks anything detailed on machine code, although it mentions that the hook codes are the same as those for microdrives. I am told that by the time you read this, there will be a new, ring-bound version of the instructions which will have provision for containing comprehensive information on machine code. The first operation is to connect everything up and load the utilities program supplied on cassette. After answering various prompts, you are ready to format a disc. Two surprises: firstly the disc is not named; *you* have to name it on the label; secondly a small code program called "systems" is also saved on the disc at the same time. A cold start involves switching on computer, inserting disc and typing RUN, when the systems program is loaded. This can be a bit irritating but does ensure that the system can be updated to cope with alterations in other systems. There is provision for one further auto-running program on any one disc.

Disciple comes with a truly comprehensive array of commands. **LOAD D1"program"**, for instance, works, as well as microdrive commands. So does **SAVE d1"program"**. **SAVE D1"program"** saves and then also gives you the short directory or catalogue, which numbers each file, tells you what it is and the start address and length if it is code. This allows you to use the very short syntax LOAD p6, when number 6 in the list is loaded from the last disc in use. All the commands one would expect are there, including wild card files (to ERASE numbers1, numbers2, numbers 3 etc., enter ERASE D1"n*"). A well thought out system.

In addition to the disc interface itself, one or two other little gizmos are incorporated. Clearly aimed at the games player is a snapshot button; this takes a copy of what is in memory, and, in this case, saves it direct to disc as a "snapshot file". This is the only type of file



which will not copy from one disc to another on a two disc system; thus piracy is minimised. The two joystick ports can be configured either as Sinclair or (left hand only) as Kempston.

A great deal of thought has been given to networking. Disciple has two network ports, each taking inexpensive standard 3.5mm jack plugs and standard cable. In this way, a shared access network can be set up with a master station and pupil stations: the master can transmit programs to the pupils and even call up their screens to check their work; they, in turn can call up programs from a central disc or print on a central printer.

Centronics

The centronics interface is fairly unremarkable. The answers to the initial utility program involve information about the printer in use if it is not an Epson compatible. The answers are incorporated in the systems program which has to be used to boost up the system; a nuisance if you are using dot matrix for rough copy and daisy wheel for the finished document. LLIST and LPRINT are supported, together with COPY SCREEN$ which copies the screen. Any control codes that you want to send to the printer are preceded by CHR$27, even if you wish to send CHR$27 itself; tortuous but effective. This can, however, be changed (it has to be for Tasword, for instance, for version 2 will transmit only four control codes at any one time).

Now for the meaty bits. I took one of my son's games and converted it onto disc. This consisted of a basic loader, a screen and a piece of machine code 38351 bytes long (virtually a full 48K Spectrum).

Loading this from tape took just short of five minutes. The same thing took a spectacular seven seconds from the Disciple! I loaded the game again and took a snapshot. Taking it took between eight and nine seconds. Reloading took five. When times get as short as this it is impossible to be very accurate; all you can say is that they are impressively fast. Saving is equally fast — I used the SAVE D1 "name" version for the picture and for the code; in each case the bulk of the seven or eight seconds was the time required to display the directory. Indeed, the only operation which took longer was formatting.

Talking of formatting, I found that there were one or two interesting comparisons to be made. Discovery has a directory length (110 files) which is independent of the disc type, and which can be extended if necessary. Disciple allows only 40 entries in single density and 80 in double. The sector length is similarly interesting, Discovery and Disciple (80 track) has 512 byte sectors. Although these bigger sectors and the lack of a compacting facility on Disciple could mean more wasteful use of the disc, this is made up for a larger formatted capacity, although do remember that you have to put the "systems" program on the disc.

My overall impressions and comments? Well, first the bad news. It's going to cost quite a lot for a complete system, for you've got the cost of a drive on top of the £79 odd for Disciple. I felt it was a bit plasticky, a bit flimsy, and I managed to get the disc connector in the wrong way round. Much of this was when, all thumbs, I was trying to connect up and get it going. Having got used to it, well, it's great. There are all sorts of additions: the snapshot, the comprehensive networking, the availability of two Sinclair or one Kempston joystick ports, the ability to run microdrive software and to have it transferred by means of the inhibit button, and above all, the incredible speed of transfer. You know, I rather fancy one myself...



Mini Studio

A Sound Sampler for any 48K or 128K Spectrum by Massimo Pilia.

To enter the program:
Type in listing 1 and save it with **SAVE "MS"LINE 5.** Then type in listing 2 and save this onto another tape with **SAVE "L2".**

Run listing 2 and when the saving prompt appears, position your FIRST tape (with MS on it) just after listing 1 and start recording. Press a key.

Rewind the tape and load listing 1. Once it has loaded and run itself you are ready to go through the following instructions.

What is Sound Sampling?

Sampling is a way of digitally recording a small amount of sound, usually for use in music. A computer's memory is an ideal medium for this type of recording, since a sound is in memory, it can be subjected to any number of bizarre alterations.

This program allows sound (up to about six seconds long) to be recorded, reversed, speeded up, slowed down and sequence-played in a specific order with other sounds. The basis of the program is the machine code that records and plays back the sound. This works by reading the cassette port at high speed, and feeding the sounds that are read into memory in the form of data bits. By reversing the process and reading the bits from memory and out of the cassette port and buzzer, the sound is reproduced. Changing the speed that the sound is played back at will also change the pitch of the sound.

There are four modes of operation:

SOUNDS — this is the mode where the sounds are sampled and cut up for use in the other modes.

PLAY — sounds set up in *SOUNDS* mode can be played back at eight pitches covering an octave.

BARS — up to eight bars, each containing up to eight sounds, can be entered in this mode.

PROGS — bars created in *BARS* mode are strung together in up to 64 steps here.

The SPACE key switches you between modes and the cursors in SOUNDS, BARS and PROGS are controlled using keys: 5=left, 6=down, 7=up and 8=right with 0 for edit.

Sounds Mode

In this mode the program is controlled with the following keys:

1 — Selects the number of beats per bar (up to eight).
2 — Selects the 'blocks' per beat (see note below).
I — Input (record a sound).
O — Output (playback).
L — Load.
S — Save.
K — Key (musical key in *PLAY* mode).

When using I, O and S they operate on the sound underlined by the cursor, this is the moving line which underlines various functions and sounds and is controlled by the keys 5, 6, 7, 8, to move it in the direction of the arrows.

The memory that sounds are sampled into is divided into 120 "blocks", each 256 bytes long. There are ten sound configurations (0-9) which are simply windows on this memory.

Each of these configurations has four values as shown on the top left of the screen. The top right displays the mode you are in. The meaning of these values is:

STA — this is the 'block' where the sound starts in memory (0-119).
LEN — the length of the sound (1-120).
DI — the direction of the sound (forwards/backwards).
DE — the delay used when playing back or recording the sound. This determines the speed and quality.

Another value — BTS — is also shown but is just the number of beats the sound occupies. This is calculated by LEN/BLOCKS/BEAT.

Recording

1. Set up the computer and tape recorder as shown in figure 1.

On most tape recorders pressing PAUSE+PLAY/REC should allow you to sample from the microphone.

2. Configure a sound so as to use up the rest of remaining memory. EG. set up sound 0 as:

N— STA —LEN— DI —DE
0— 000 —120—FO—01

by moving the cursor with keys 5-8 until it is beneath the value you wish to change, press 0 and then enter the value you require.

The delay is set to 01 for maximum quality. If the delay was higher the sound could then be replayed at higher pitches (ie. lower delays) but would be of poorer quality.

3. Press I for input and reply "y" to the "sure?" prompt.

4. Press a key to start sampling when you are ready. The sample is complete when the "PRESS A KEY" prompt disappears.

5. Press O to listen to your sample. It will sound best through headphones plugged into the ear socket at the back of the computer.

6. Now you must tighten up the sound, ie. chop off any excess sound before or after the part you want. Start by increasing the start value (STA) bit by bit, you will have to decrease the LEN value before each increase of STA as the TOTAL of STA+LEN is a maximum of 120. Listen to the sound each time until there is no superfluous sound before the portion you want.

Now decrease the length (LEN) value until the sample is exactly the portion of sound you require.

7. Your sample is now complete. You may reverse it by moving the cursor to the DI column and pressing 0. You may also change the delay (speed). I shall explain the various ways of using a sample in a moment.

Play Mode

This mode uses the following keys:

1 — Record.
2 — Playback.

In the Record sub-mode the sound currently underlined by the cursor in *SOUNDS* is turned into a musical (or not so

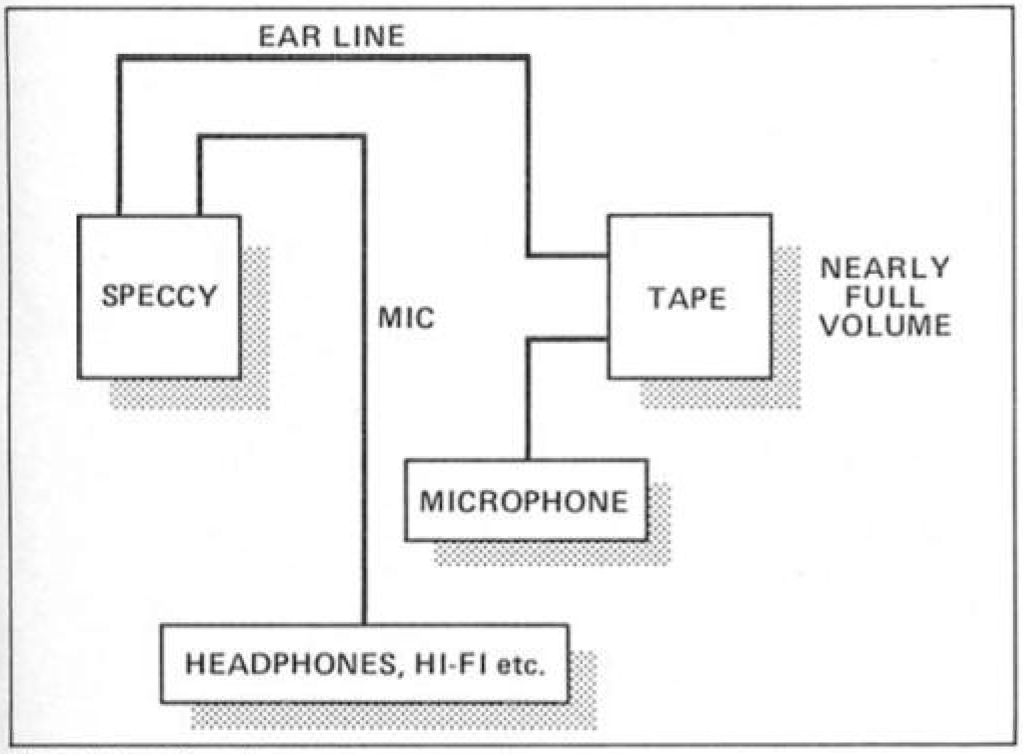


Figure 1. Sampling set-up.

musical) instrument played on keys 1-8 in the key specified by the K key from the *SOUNDS* mode.

0 toggles recording of tunes ON/OFF. 9 exits the sub-mode.

In the Playback sub-mode a tune recorded in Record can be replayed by tapping out the rhythm on the 0 key. 9 exits.

Bars Mode

This mode uses the following keys:

O — Run/Initialise (bars).
E — Toggle Stop/Repeat.
C — Toggle beat counter ON/OFF

Hold down any key on the bottom row to stop the sequence looping endlessly (this happens when repeat is ON).

When the beat counter is ON, any bar entered that does not contain the right number of beats (ie. when the total BTS value of every sound in the bar is not the same as BEATS/BAR) will be rejected. This bar must then either be entered correctly or deleted by just pressing ENTER.

In this mode eight bars (lists of sounds) can be edited and stored. These can be played back in order in this mode or in a more complicated sequence from *PROGS* mode.

Entering bars:
1. Move the cursor under the bar to be edited and press 0.
2. Type in your bar. (Eg. 11134132). The numbers are the numbers of the sound samples that you set up in *SOUNDS* mode.
3. Enter any another bars you want to enter.
4. Press O and then enter "i" to initialise.
5. Press O and then "r" for RUN to listen to your stroke of musical genius.

The example bar in step 2 will play sounds 1,1,1,3,4,1,3,2 in that order. When counting the beats it counts sound 1 four times and 3 twice. Bars are split into eight parts. I prefer to leave the counter OFF but it is there for a purpose.

Progs Mode

This mode is controlled by keys O and E:

O — Run/Initialise (sequence).
E — Toggle STOP/REPEAT.

Hold down any key on the bottom row to stop a sequence from playing.

Bars can be strung together in up to 64 steps. This is for longer, more complicated sequences and for quicker editing.

Entering steps:
1. Position the cursor under the value to be edited.
2. Press 0 to edit and enter the number of the bar you want. Enter 0 to signify the end of the sequence.
3. Enter any other steps you want to enter.
4. Press O and then I to initialise. Press O and then "r" to listen to the sequence.

Whenever an alteration is made in either *PROGS* or *BARS* mode remember to re-initialise them.

Using a Sample

Once a sample has been created several things can be done with it:

The sample can be used as a musical instrument in *PLAY* mode.
2. The sound can be put into a simpler sequence on its own. If you have sampled some speech as: 0-000-016-FO-01 then you could set up sounds 1 and 2 as shorter versions of 0:—
1-000-002-FO-01
2-000-004-FO-01

In *BARS* mode enter bar 1 as: 211220. Initialise and run it for a N-N-N-Nineteen effect.

3. Echo can be simulated by "layering" the sound. Set up the sounds like this:

```
N— STA —LEN— DI —DE
0— 000 —004— FO—01
1— 002 —004— FO—01
2— 004 —004— FO—01
3— 006 —004— FO—01
4— 008 —004— FO—01
5— 010 —004— FO—01
6— 012 —004— FO—01
7— 014 —004— FO—01
```

Enter bar 1 as 01234567. Initialise and run it.

4. The sound may also be used as part of a more complicated sequence using other sounds (ie. a drum beat).

Sequencing a Tune

Sequencing a tune poses a problem because when the sounds pitch is changed its length in time is also changed. This means that the rhythm goes out the window. As the program doesn't have a timer another method has to be used:

Set up sounds 0-7 thus:

```
N— STA —LEN— DI —DE
0— 000 —008— FO—00
1— 008 —008— FO—08
2— 016 —008— FO—16
3— 024 —008— FO—20
4— 032 —008— FO—28
5— 040 —008— FO—36
6— 048 —008— FO—44
7— 056 —008— FO—48
```

Sample the same sound (presumably a recorded sound) into each area. Now change all delays back to 00.

These delay settings are in the key of C (at a guess) and represent the notes C,D,E,F,G,A,B and C. Add 4 to a delay for a flat and subtract 4 for a sharp.

Example Rhythm

If you managed to take all that in you are ready for your first drum rhythm using MINI-STUDIO.

1. Enter SOUNDS mode.
2. Set up sounds 1-3 like this:

1-000-003-FO-01
2-003-003-FO-01
3-006-003-FO-01

3. Input or load a bass drum sound into 1.
4. Input or load a snare drum sound into 2.
5. Input a blank (silence) into 3.
6. Enter BARS mode.
7. Turn the beat counter off (press C).
8. Enter the following bars:

13132313 13332333 13232322 13132133
33132133 11112111 33112111 13222223

9. Initialise.
10. Enter progs mode and enter these bar steps:
1 2 1 3 4 2 5 3 6 2 7 3 6 8 7 3
11. Initialise and run this.

```
N: -STA-LEN-DI-DE-BTS  MODE: SOUNDS
0: -000-004-FO-01/1
1: -000-004-FO-01/1            5/6/7/8
2: -000-004-FO-01/1              FOR
3: -000-004-FO-01/1            CURSORS
4: -000-004-FO-01/1
5: -000-004-FO-01/1             SPACE
6: -000-004-FO-01/1              FOR
7: -000-004-FO-01/1             MODES
8: -000-004-FO-01/1
9: -000-004-FO-01/1
1-B/BAR:8  I=INPUT   L=LOAD  K=KEY
2-BLK/B:4  O=OUTPUT  S=SAVE   (C)



C=TOGGLE COUNT (ON)  STUDIO ©1986
E=TOGGLE END (STOP)       M.Pilia
   O=RUN/INITIALISE SEQUENCES
00000000000000000000000000000000
00000000000000000000000000000000
```

Figure 2. Example of the screen display.

Listing 1

```
   2 REM MINI-STUDIO
   5 CLEAR VAL "33791": LOAD ""C
ODE
  10 LET k=VAL "8": LET k$="2212
221": LET l$="CDEFGAB": LET d=PI
-PI: LET s=PI-PI: LET px=VAL "20
": LET py=PI-PI: DIM c$(VAL "64"
): DIM d(VAL "8",VAL "2"): LET t
=PI-PI: LET s$="n": LET f1=PI-PI
: LET bar=VAL "8": LET blk=VAL "
4": LET x1=VAL "14": LET y1=PI-P
I: LET f=PI-PI: DIM s(VAL "10",V
AL "4"): DIM p$(VAL "8",VAL "8")
: LET x=PI/PI: LET y=INT PI
  20 FOR a=PI/PI TO VAL "10": LE
T s(a,VAL "2")=VAL "4": LET s(a,
VAL "4")=1: NEXT a: FOR a=PI/PI
TO VAL "64": LET c$(a)="0": NEXT
 a
  32 GO SUB VAL "950"
  33 CLS : PRINT AT 17,20;"STUDI
O1986";AT 18,23;"M.Pilia"
  35 PRINT AT PI-PI,PI-PI;"N:-ST
A-LEN-DI-DE-BTS"
  36 PRINT  OVER PI/PI;AT PI-PI,
PI-PI;"_  ___ ___ __ __ ___"
  37 PRINT  PAPER VAL "6";AT VAL
 "11",PI-PI;"1-B/BAR:";bar;" I=I
NPUT  L=LOAD K=KEY 2-BLK/B:";blk
;" O=OUTPUT S=SAVE  (";l$(k);")
 ";AT VAL "2",VAL "24";"5/6/7/8"
;AT INT PI,VAL "24";"  FOR  ";AT
 VAL "4",VAL "24";"CURSORS";AT V
AL "6",VAL "24";" SPACE ";AT VAL
 "7",VAL "24";"  FOR  ";AT VAL "
8",VAL "24";" MODES ";AT VAL "19
",PI-PI;"   O=RUN/INITIALISE SEQ
UENCES   "
  40 PRINT AT PI/PI,PI-PI;: FOR
a=PI-PI TO VAL "9": PRINT AT a+P
I/PI,PI-PI; PAPER VAL "6"; INK V
AL "2";a;: PRINT ":-";: IF s(a+P
I/PI,PI/PI)<VAL "10" THEN  PRINT
 PI-PI;
  45 IF s(a+PI/PI,PI/PI)<VAL "10
0" THEN  PRINT PI-PI;
  50 PRINT s(a+PI/PI,PI/PI);"-";
: IF s(a+PI/PI,VAL "2")<VAL "10"
 THEN  PRINT PI-PI;
  51 IF s(a+PI/PI,VAL "2")<VAL "
100" THEN  PRINT PI-PI;
  52 PRINT s(a+PI/PI,VAL "2");
  55 IF s(a+PI/PI,INT PI)=PI/PI
THEN  PRINT "-BA-";
  57 IF s(a+PI/PI,INT PI)=PI-PI
THEN  PRINT "-FO-";
  58 IF s(a+PI/PI,VAL "4")<VAL "
10" THEN  PRINT PI-PI;
  59 PRINT s(a+PI/PI,VAL "4");
  60 PRINT "/";s(a+PI/PI,VAL "2"
)/blk;: LET b$=STR$ (s(a+PI/PI,V
AL "2")/blk): GO SUB VAL "197":
NEXT a
  65 IF f1=PI/PI THEN  LET f1=PI
-PI: PRINT  OVER PI/PI;AT x,y;"_
": RETURN
  70 PRINT AT VAL "13",PI-PI;: F
OR a=PI/PI TO VAL "8"
  80 IF a/VAL "2"=INT (a/VAL "2"
) THEN  PRINT p$(a);
  90 IF a/VAL "2"<>INT (a/VAL "2
") THEN  PRINT  PAPER VAL "5";p$
(a);
  92 IF a/VAL "4"=INT (a/VAL "4"
) THEN  PRINT : PRINT
  93 NEXT a
  94 IF f=PI/PI THEN  RETURN
  96 BEEP VAL ".1",VAL "12": PRI
NT AT PI-PI,VAL "21";"MODE:"; PA
PER VAL "6"; INK VAL "2";"SOUNDS
": GO SUB VAL "160"
 110 IF INKEY$="8" AND y=VAL "11
" THEN  GO SUB VAL "160": LET y=
VAL "14": GO SUB VAL "160"
 115 IF INKEY$="8" AND y<VAL "11
" THEN  GO SUB VAL "160": LET y=
y+VAL "4": GO SUB VAL "160"
 120 IF INKEY$="5" AND y=VAL "14
" THEN  GO SUB VAL "160": LET y=
VAL "11": GO SUB VAL "160"
 125 IF INKEY$="5" AND y<VAL "14
" AND y>INT PI THEN  GO SUB VAL
"160": LET y=y-VAL "4": GO SUB V
AL "160"
 130 IF INKEY$="6" AND x<VAL "10
" THEN  GO SUB VAL "160": LET x=
x+PI/PI: GO SUB VAL "160"
 132 IF INKEY$="1" THEN  GO SUB
VAL "500"
 135 IF INKEY$="2" THEN  GO SUB
VAL "510"
 137 IF INKEY$="s" THEN  GO SUB
VAL "900"
 138 IF INKEY$="l" THEN  GO TO V
AL "910"
 139 IF INKEY$="k" THEN  GO SUB
VAL "950"
 140 IF INKEY$="7" AND x>PI/PI T
HEN  GO SUB VAL "160": LET x=x-P
I/PI: GO SUB VAL "160"
 145 IF INKEY$="0" THEN  GO SUB
VAL "170"
 147 IF INKEY$=" " THEN  GO TO V
AL "200"
 148 IF INKEY$="i" THEN  GO SUB
VAL "600"
 149 IF INKEY$="o" THEN  GO SUB
VAL "608"
 150 GO TO VAL "110"
 151 IF s(x,INT PI)=PI-PI THEN
PRINT AT x,y;"FO";AT x,INT PI;:
LET b$=STR$ s(x,PI/PI): FOR d=PI
/PI TO INT PI-LEN b$: PRINT "0";
: NEXT d: PRINT s(x,PI/PI): GO S
UB VAL "160": RETURN
 152 IF s(x,INT PI)=PI/PI THEN
PRINT AT x,y;"BA";AT x,INT PI;:
LET b$=STR$ s(x,PI/PI): FOR d=PI
/PI TO INT PI-LEN b$: PRINT "0";
: NEXT d: PRINT s(x,PI/PI): GO S
UB VAL "160": RETURN
 160 PRINT AT x,y; OVER PI/PI;"_
": BEEP VAL ".01",VAL "12": RETU
RN
 170 IF y=INT PI THEN  LET b=PI/
PI
 175 IF y=VAL "7" THEN  LET b=VA
```

```
L "2"
 176 IF y=VAL "11" AND s(x,INT P
I)=PI-PI THEN  LET s(x,PI/PI)=s(
x,PI/PI)+s(x,VAL "2"): LET s(x,I
NT PI)=PI/PI: GO SUB VAL "151":
RETURN
 177 IF y=VAL "11" AND s(x,INT P
I)=PI/PI THEN  LET s(x,PI/PI)=s(
x,PI/PI)-s(x,VAL "2"): LET s(x,I
NT PI)=PI-PI: GO SUB VAL "151":
RETURN
 178 IF y=VAL "14" THEN  LET b=V
AL "4"
 179 BEEP VAL ".01",VAL "12": IN
PUT ":";s(x,b): PRINT AT x,y;: I
F b=VAL "2" AND s(x,b)>VAL "120"
 THEN  GO TO VAL "179"
 180 IF b=VAL "4" AND s(x,b)>VAL
 "64" OR b=VAL "4" AND s(x,b)<PI
-PI THEN  GO TO VAL "179"
 181 IF s(x,INT PI)=PI/PI THEN
IF s(x,PI/PI)-s(x,VAL "2")<PI-PI
 THEN  GO TO VAL "179"
 182 IF s(x,INT PI)=PI-PI THEN
IF s(x,PI/PI)+s(x,VAL "2")>VAL "
120" THEN  GO TO VAL "179"
 183 IF s(x,b)<VAL "10" THEN  PR
INT PI-PI;
 184 IF b=VAL "2" AND s(x,VAL "2
")<PI/PI THEN  GO TO VAL "179"
 186 IF b=PI/PI AND s(x,b)<VAL "
100" OR b=VAL "2" AND s(x,b)<VAL
 "100" THEN  PRINT PI-PI;
 187 IF b=VAL "4" THEN  PRINT s(
x,b)
 189 IF b=VAL "4" THEN  GO SUB V
AL "160": RETURN
 190 IF b=VAL "2" THEN  PRINT s(
x,VAL "2"): PRINT AT x,VAL "17";
s(x,VAL "2")/blk;: GO SUB VAL "1
96": GO SUB VAL "160": RETURN
 195 PRINT s(x,b): GO SUB VAL "1
60": RETURN
 196 LET b$=STR$ (s(x,VAL "2")/b
lk)
 197 FOR c=PI/PI TO VAL "6"-LEN
b$: PRINT " ";: NEXT c: RETURN
 199 RETURN
 200 BEEP VAL ".1",VAL "12": PRI
NT  PAPER VAL "6"; INK VAL "2";A
T PI-PI,VAL "26";"PLAY  ": GO SU
B VAL "600"
 202 PRINT  PAPER VAL "6"; INK P
I-PI;AT VAL "20",VAL "0";"1-PLAY
/RECORD                    ";AT V
AL "21",VAL "0";"2-PLAYBACK
                    ": PAUSE PI/PI:
 PAUSE PI-PI: LET i$=INKEY$
 205 IF i$="1" THEN  PRINT  PAPE
R VAL "6"; INK PI-PI;AT VAL "20"
,PI-PI;"        1-8=NOTES 9=EXIT
        ";AT VAL "21",PI-PI;"
  0=TOGGLE RECORD ON/OFF       ":
LET a=USR VAL "65300": IF a<VAL
"65000" THEN  POKE a,PI-PI
 206 IF i$="1" THEN  GO TO VAL "
202"
 210 IF i$="2" THEN  PRINT  PAPE
R VAL "6"; INK PI-PI;AT VAL "20"
,PI-PI;"        0=PLAYBACK NOTE
        ";AT VAL "21",PI-PI;"
   9=EXIT PLAYBACK MODE        ":
RANDOMIZE USR VAL "65450": GO TO
 VAL "202"
 220 PRINT AT VAL "20",PI-PI,,,,
 300 BEEP VAL ".1",VAL "12": PRI
NT  PAPER VAL "6"; INK VAL "2";A
T PI-PI,VAL "26";"BARS"
 305 IF t=PI-PI THEN  PRINT AT V
AL "17",PI-PI;"C=TOGGLE COUNT (O
N) "
 307 IF t=PI/PI THEN  PRINT AT V
AL "17",PI-PI;"C=TOGGLE COUNT (O
FF)"
 308 IF s=PI-PI THEN  PRINT AT V
AL "18",PI-PI;"E=TOGGLE END (STO
P)   "
 309 IF s=PI/PI THEN  PRINT AT V
AL "18",PI-PI;"E=TOGGLE END (REP
EAT)"
 310 BEEP VAL ".01",VAL "12": PR
INT AT x1,y1;"▀▀▀▀▀▀▀▀"
 320 IF INKEY$="8" AND y1<VAL "2
3" THEN  PRINT AT x1,y1;"
 ": LET y1=y1+VAL "8": GO TO VAL
 "310"
 330 IF INKEY$="5" AND y1>PI-PI
THEN  PRINT AT x1,y1;"         ":
 LET y1=y1-VAL "8": GO TO VAL "3
10"
 340 IF INKEY$="7" AND x1>VAL "1
5" THEN  PRINT AT x1,y1;"
 ": LET x1=x1-VAL "2": GO TO VAL
 "310"
 350 IF INKEY$="6" AND x1<VAL "1
5" THEN  PRINT AT x1,y1;"
 ": LET x1=x1+VAL "2": GO TO VAL
 "310"
 360 IF INKEY$=" " THEN  GO TO V
AL "800"
 362 IF INKEY$="e" THEN  LET s=N
OT s: GO TO VAL "300"
 365 IF INKEY$="o" THEN  GO SUB
VAL "700"
 367 IF INKEY$="c" THEN  LET t=N
OT t: GO TO VAL "305"
 370 IF INKEY$="0" OR f=VAL "2"
THEN  INPUT  LINE p$((((x1-VAL "
11")*VAL "2")+y1/VAL "8")-VAL "5
"): LET f=PI/PI: GO SUB VAL "550
": GO SUB VAL "70": LET f=PI-PI
 400 GO TO VAL "320"
 500 INPUT "B/BAR(1-8):";bar
 502 IF bar<PI/PI OR bar>VAL "8"
 THEN  GO TO VAL "500"
 504 BEEP VAL ".01",VAL "12": PR
INT  PAPER VAL "6";AT VAL "11",V
AL "8";bar: RETURN
 510 INPUT "BLK/B:";blk
 512 IF blk<>PI/PI AND blk<>VAL
"2" AND blk<>VAL "4" AND blk<>VA
L "8" THEN  GO TO VAL "510"
 514 PRINT  PAPER VAL "6";AT VAL
 "12",VAL "8";blk: LET f1=PI/PI:
 GO SUB VAL "40": BEEP VAL ".01"
,VAL "12": RETURN
 550 IF t=PI/PI THEN  LET b=bar:
 GO TO VAL "580"
 560 IF p$((((x1-VAL "11")*VAL "
2")+y1/VAL "8")-VAL "5")="
  " THEN  GO TO VAL "580"
 561 LET b=PI-PI: FOR a=1 TO 8:
IF p$((((x1-11)*2)+y1/8)-5,a)="
" THEN  NEXT a
 562 IF a=9 THEN  GO TO VAL "570
"
 563 IF p$((((x1-11)*2)+y1/8)-5,
a)="s" THEN  GO TO VAL "580"
 565 LET b=b+s((VAL p$((((x1-11)
*2)+y1/8)-5,a))+1,2)/blk: NEXT a
 570 IF b<>bar THEN  BEEP VAL ".
1",VAL "-12": LET f=PI/PI: GO SU
B VAL "70": LET f=VAL "2": GO TO
 VAL "370"
 580 BEEP VAL ".01",VAL "12": RE
TURN
 600 INPUT "SURE?"; LINE s$: IF
s$<>"y" THEN  RETURN
 601 IF s(x,INT PI)=PI/PI THEN
POKE VAL "65010",VAL "43"
 602 IF s(x,INT PI)=PI-PI THEN
POKE VAL "65010",VAL "35"
 603 IF s(x,VAL "4")=PI-PI THEN
 FOR a=VAL "65030" TO VAL "65033
": POKE a,PI-PI: NEXT a: POKE VA
L "65003",VAL "133"+s(x,PI/PI):
POKE VAL "65005",s(x,VAL "2"): G
O TO 607
 605 IF s(x,VAL "4")>PI-PI THEN
 POKE VAL "65030",VAL "6": POKE
VAL "65032",VAL "16": POKE VAL "
65033",VAL "254": POKE VAL "6500
3",VAL "133"+s(x,PI/PI): POKE VA
L "65005",s(x,VAL "2"): POKE VAL
 "65031",s(x,VAL "4")
 607 BEEP VAL ".01",VAL "12": PR
INT AT VAL "21",PI-PI; PAPER VAL
 "6";"             PRESS A KEY
      ": PAUSE PI-PI: RANDOMIZE
USR VAL "65000": PRINT AT VAL "2
1",PI-PI;,,: RETURN
 608 IF s(x,INT PI)=PI/PI THEN
POKE VAL "65067",VAL "43"
 609 IF s(x,INT PI)=PI-PI THEN
POKE VAL "65067",VAL "35"
 610 IF s(x,VAL "4")=PI-PI THEN
 FOR h=VAL "65089" TO VAL "65092
": POKE h,PI-PI: NEXT h: POKE VA
L "65060",VAL "133"+s(x,PI/PI):
POKE VAL "65062";s(x,VAL "2"): R
ANDOMIZE USR VAL "65040": RETURN

 615 IF s(x,VAL "4")>PI-PI THEN
 POKE VAL "65089",VAL "6": POKE
VAL "65091",VAL "16": POKE VAL "
65092",VAL "254": POKE VAL "6506
3",VAL "133"+s(x,PI/PI): POKE VA
L "65062",s(x,VAL "2"): POKE VAL
 "65090",s(x,VAL "4"): RANDOMIZE
 USR VAL "65040": RETURN
 700 INPUT "R or I?"; LINE i$
 702 POKE VAL "65119",d+(s=PI/PI
)
 705 IF i$="r" THEN : PRINT  PAP
ER 6;AT 21,0;"           PRESS A
 KEY           ": OUT VAL "254",V
AL "31": POKE VAL "65116",VAL "2
55": POKE VAL "65117",VAL "90":
PAUSE PI-PI: RANDOMIZE USR VAL "
65115": PRINT AT VAL "21",PI-PI,
,: RETURN
 707 IF i$<>"i" THEN  RETURN
 710 LET sp=PI-PI: LET c=VAL "23
296": FOR a=PI/PI TO VAL "8": LE
T d(a,PI/PI)=c-VAL "23295": FOR
b=PI/PI TO VAL "8"
 711 IF p$(a,b)=" " OR p$(a,b)="
s" THEN  LET sp=sp+PI/PI: GO TO
VAL "745"
 712 LET d=((a-1)*8)+b-sp
 715 IF s(VAL p$(a,b)+1,3)=0 THE
N  POKE c,s(VAL p$(a,b)+1,1)
 720 IF s(VAL p$(a,b)+1,3)=1 THE
N  POKE c,s(VAL p$(a,b)+1,1)+133
 725 LET c=c+1
 730 POKE c,s(VAL p$(a,b)+1,2)
 735 LET c=c+1
 737 POKE c,s(VAL p$(a,b)+1,4)
 738 IF s(VAL p$(a,b)+1,4)<>0 TH
EN  POKE c,(PEEK c)+128
 739 LET c=c+1
 745 NEXT b: NEXT a
 746 IF s=PI-PI THEN  POKE VAL "
65119",d
 747 IF s=PI/PI THEN  POKE VAL "
65119",d+PI/PI: POKE c+PI/PI,VAL
 "128"
 750 IF a=VAL "9" THEN  LET a=VA
L "8"
 755 FOR a=PI/PI TO VAL "7": LET
 d(a,VAL "2")=d(a+PI/PI,PI/PI)-d
(a,PI/PI): NEXT a
 756 LET d(VAL "8",VAL "2")=(c-V
AL "23295")-d(VAL "7",PI/PI)-d(V
AL "7",VAL "2")
 760 BEEP VAL ".01",VAL "12": RE
TURN
 800 BEEP VAL ".1",VAL "12": PRI
NT  PAPER VAL "6"; INK VAL "2";A
T PI-PI,VAL "26";"PROGS"
 802 PRINT AT VAL "20",PI-PI;c$(
 TO VAL "32");AT VAL "21",PI-PI;
```

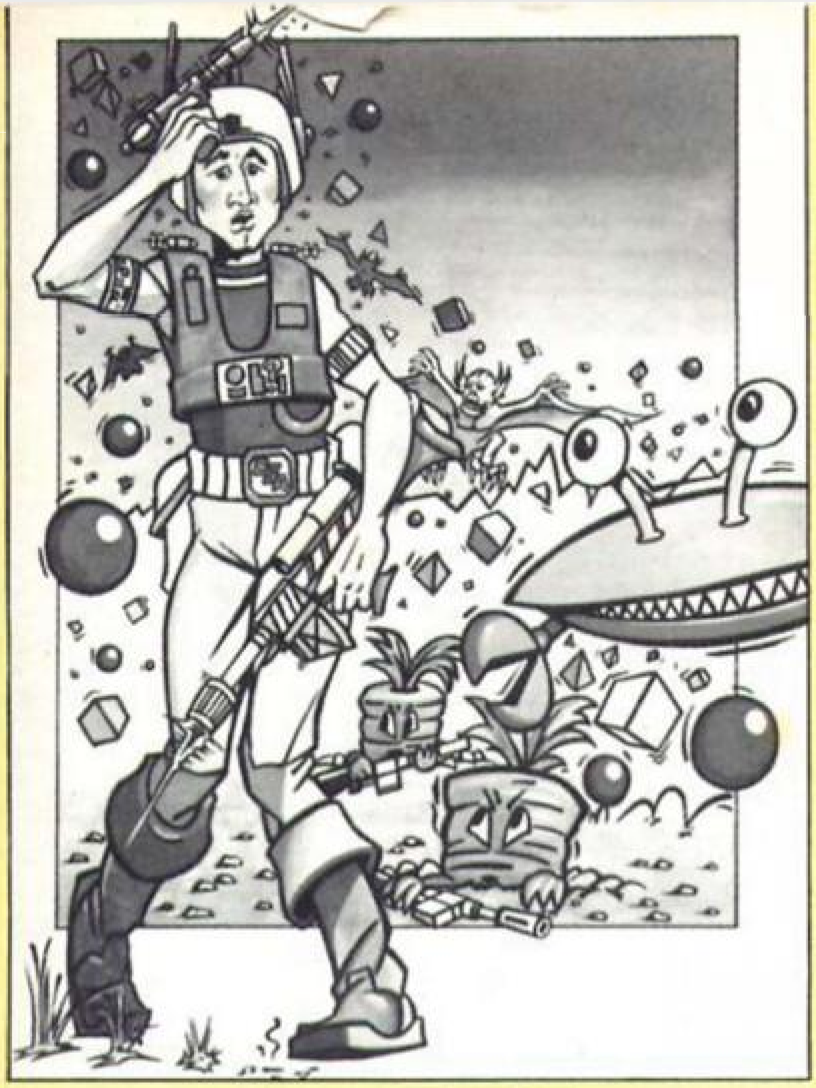
```
c$(VAL "33" TO )
 805 GO SUB VAL "899"
 810 IF INKEY$="6" AND px=VAL "2
0" THEN  GO SUB VAL "899": LET p
x=VAL "21": GO SUB VAL "899"
 812 IF INKEY$="7" AND px=VAL "2
1" THEN  GO SUB VAL "899": LET p
x=VAL "20": GO SUB VAL "899"
 815 IF INKEY$="8" AND py<VAL "3
1" THEN  GO SUB VAL "899": LET p
y=py+PI/PI: GO SUB VAL "899"
 816 IF INKEY$="5" AND py>PI-PI
THEN  GO SUB VAL "899": LET py=p
y-PI/PI: GO SUB VAL "899"
 817 IF INKEY$="0" THEN  GO SUB
VAL "860"
 818 IF INKEY$=" " THEN  PRINT A
T x,y; OVER PI/PI;"_": GO TO VAL
 "96"
 819 IF INKEY$="o" THEN  GO SUB
VAL "830"
 820 IF s=PI/PI AND INKEY$="e" T
HEN  LET s=PI-PI: PRINT AT VAL "
18",PI-PI;"E=TOGGLE END (STOP)
": BEEP VAL ".01",VAL "12"
 821 IF s=PI-PI AND INKEY$="e" T
HEN  LET s=PI/PI: PRINT AT VAL "
18",PI-PI;"E=TOGGLE END (REPEAT)
": BEEP VAL ".01",VAL "12"
 825 GO TO VAL "810"
 830 INPUT "R or I?"; LINE i$: I
F i$="r" THEN  OUT VAL "254",VAL
 "31": PRINT  PAPER VAL "6";AT V
AL "21",PI-PI;"              PRESS A
 KEY              ": PAUSE PI-PI: R
ANDOMIZE USR VAL "65250": PRINT
AT VAL "21",PI-PI,,: PRINT AT VA
L "21",PI-PI;c$(VAL "33" TO ): I
F px=VAL "21" THEN  PRINT AT px,
py; OVER PI/PI;"_"
 835 IF i$<>"i" THEN  RETURN
 840 LET c=PI/PI: FOR a=VAL "337
92" TO VAL "33919" STEP VAL "2"
 842 IF c$(c)="0" THEN  GO TO 85
1
 845 POKE a,d(VAL c$(c),1)-1
 850 POKE a+1,d(VAL c$(c),2)/3:
LET c=c+1: NEXT a
 851 IF s=PI-PI THEN  POKE a,VAL
 "255"
 852 IF s=PI/PI THEN  POKE a,VAL
 "224"
 855 BEEP VAL ".01",VAL "12": RE
TURN
 860 INPUT  LINE c$((px-VAL "20"
)*VAL "32"+py+PI/PI)
 865 LET a=CODE c$((px-VAL "20")
*VAL "32"+py+PI/PI)
 867 IF a<VAL "48" OR a>VAL "56"
 THEN  GO TO VAL "860"
 870 GO TO VAL "800"
 899 PRINT AT px,py; OVER PI/PI;
"_": BEEP VAL ".01",VAL "12": RE
TURN
 901 INPUT "Filename?"; LINE i$:
 IF LEN i$<PI/PI OR LEN i$>VAL "
10" THEN  GO TO VAL "901"
 902 IF i$="r" THEN  GO TO VAL "
140"
 903 SAVE i$CODE VAL "34048"+s(x
,PI/PI)*VAL "256",s(x,VAL "2")*V
AL "256"
 904 INPUT "Verify(y/n)?"; LINE
s$: IF s$<>"y" THEN  RETURN
 905 VERIFY i$CODE
 906 GO SUB VAL "33"
 910 INPUT "Filename?"; LINE i$:
 IF LEN i$>VAL "10" THEN  GO TO
VAL "910"
 911 IF i$="r" THEN  GO TO VAL "
140"
 912 INPUT "Start?";i: IF i>VAL
"119" THEN  GO TO VAL "912"
 913 PRINT AT VAL "18",PI-PI;
 914 LOAD i$CODE VAL "34048"+i*V
AL "256"
 915 GO TO VAL "33"
 950 IF k<>VAL "8" THEN  LET k=k
+PI/PI
 955 IF k=VAL "8" THEN  LET k=PI
/PI
 960 LET t$=k$(k TO )+k$(VAL "2"
 TO k): POKE VAL "65450",VAL "49
": FOR a=VAL "65451" TO VAL "654
56": POKE a,PEEK (a-PI/PI)-VAL "
4"*(VAL t$(a-VAL "65450")): NEXT
 a: POKE VAL "65457",PI/PI: PRIN
T AT VAL "12",VAL "20"; PAPER 6;
i$(k): BEEP VAL ".01",VAL "12":
RETURN
```

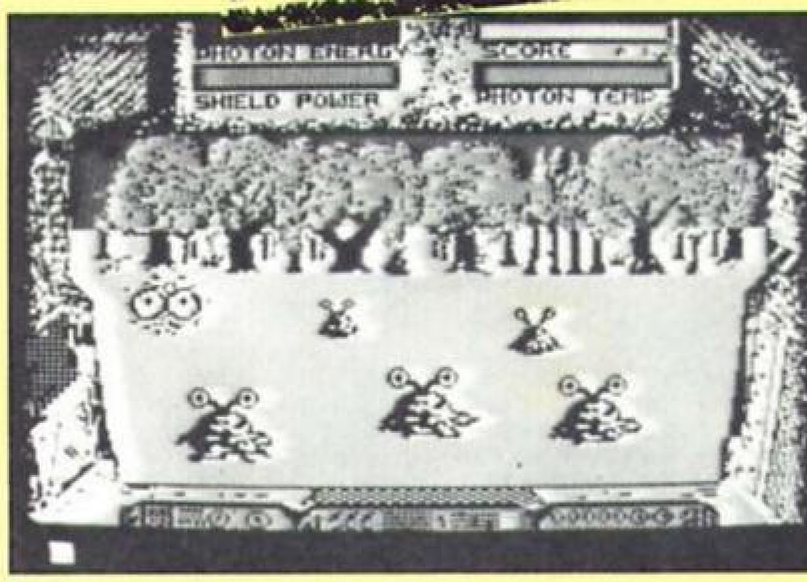
Listing 2

```
   5 REM Mini Studio MC LOADER
  10 CLEAR 4e4: FOR a=65000 TO 6
5488 STEP 8: LET d=0: FOR b=a TO
 a+7: READ c: LET d=d+c: POKE b,
c: NEXT b: READ c: IF d<>c THEN
 GO TO 50
  15 NEXT a
  20 PRINT "SAVING"
  30 SAVE "MSCODE"CODE 65000,496
  40 STOP
  50 PRINT "THERE IS AN ERROR IN
 THE CODE,  PLEASE CHECK DATA"
  60 STOP
 100 DATA 243,33,0,133,6,4,197,6
,622
 110 DATA 0,197,35,6,8,219,254,2
03,922
 120 DATA 119,40,4,203,195,24,2,
203,790
 130 DATA 131,203,11,16,240,115,
6,1,723
 140 DATA 16,254,193,16,228,193,
16,222,1138
 150 DATA 251,201,0,0,0,0,0,0,45
2
 160 DATA 243,62,231,219,254,230
,31,238,1508
 170 DATA 31,87,33,0,133,6,4,197
,491
 180 DATA 6,0,197,35,94,6,8,203,
549
 190 DATA 67,40,6,62,31,211,254,
24,695
 200 DATA 4,62,7,211,254,203,11,
16,768
 210 DATA 238,6,0,16,254,193,16,
226,949
 220 DATA 62,231,219,254,230,31,
238,31,1296
 230 DATA 146,32,5,193,16,209,25
1,201,1053
 240 DATA 193,251,201,33,255,90,
6,65,1094
 250 DATA 35,24,96,126,87,62,43,
203,676
 260 DATA 122,32,8,62,35,95,122,
198,674
 270 DATA 133,87,123,50,43,254,1
22,50,862
 280 DATA 36,254,35,126,87,62,31
,50,681
 290 DATA 52,254,203,122,40,5,62
,7,745
 300 DATA 50,52,254,122,50,38,25
4,35,855
 310 DATA 126,203,127,32,14,50,6
5,254,871
 320 DATA 50,66,254,50,67,254,50
,68,859
 330 DATA 254,24,20,203,191,50,6
6,254,1062
 340 DATA 62,6,50,65,254,62,16,5
0,565
 350 DATA 67,254,62,254,50,68,25
4,229,1238
 360 DATA 197,213,205,24,254,209
,193,225,1520
 370 DATA 16,158,201,35,94,203,1
23,43,873
 380 DATA 40,153,24,4,205,91,254
,201,972
 390 DATA 62,62,219,254,230,31,2
38,31,1127
 400 DATA 40,242,201,0,0,0,0,0,4
83
 410 DATA 0,0,33,0,132,126,22,0,
313
 420 DATA 95,1,255,90,235,9,214,
224,1123
 430 DATA 40,240,214,31,200,34,9
2,254,1105
 440 DATA 235,35,126,50,95,254,2
29,205,1229
 450 DATA 91,254,225,35,62,62,21
9,254,1202
 460 DATA 230,31,238,31,192,24,2
14,0,960
 470 DATA 0,0,0,0,243,22,0,62,32
7
 480 DATA 247,219,254,203,71,40,
42,203,1279
 490 DATA 79,40,43,203,87,40,44,
203,739
 500 DATA 95,40,45,203,103,40,46
,62,634
 510 DATA 239,219,254,203,103,40
,43,203,1304
 520 DATA 95,40,44,203,87,40,45,
203,757
 530 DATA 79,40,62,203,71,40,60,
24,579
 540 DATA 206,58,170,255,24,33,5
8,171,975
 550 DATA 255,24,28,58,172,255,2
4,23,839
 560 DATA 58,173,255,24,18,58,17
4,255,1015
 570 DATA 24,13,58,175,255,24,8,
58,615
 580 DATA 176,255,24,3,58,177,25
5,50,998
 590 DATA 66,254,245,213,197,205
,24,254,1458
 600 DATA 193,209,241,203,66,40,
152,24,1128
 610 DATA 25,251,201,203,66,40,4
,22,812
 620 DATA 0,24,4,14,1,22,255,62,
382
 630 DATA 239,219,254,203,71,40,
248,195,1469
 640 DATA 23,255,6,253,2,12,62,2
31,844
 650 DATA 185,40,3,195,23,255,22
,0,723
 660 DATA 24,249,49,41,33,29,21,
13,459
 670 DATA 5,1,6,253,14,1,62,239,
581
 680 DATA 219,254,203,79,200,203
,71,32,1261
 690 DATA 245,10,50,66,254,214,0
,200,1039
 700 DATA 197,205,24,254,193,12,
121,214,1220
 710 DATA 231,200,24,226,0,0,0,0
,681
```

ABSORBER THE GREAT!

COMPETITION





There are thirty copies of Martech's Cosmic Shock Absorber to be won.

Cosmic Shock Absorber is a would-be Superhero with aspirations to join the Supermans and Batmans of this world. Unfortunately he must serve his apprenticeship by taking on foes such as rampaging carrots armed with machine guns and legions of bouncing balls. Cosmic Shock Absorber may be lacking super-strength and superabilities, but with your help he can make it into the Superhero class.

There are thirty copies of Martech's off-beat 3-D shoot 'em up to be won, and all you have to do is name the superheroes who fought the following three supervillains.

a) The Penguin
b) Lex Luthor
c) Emperor Ming

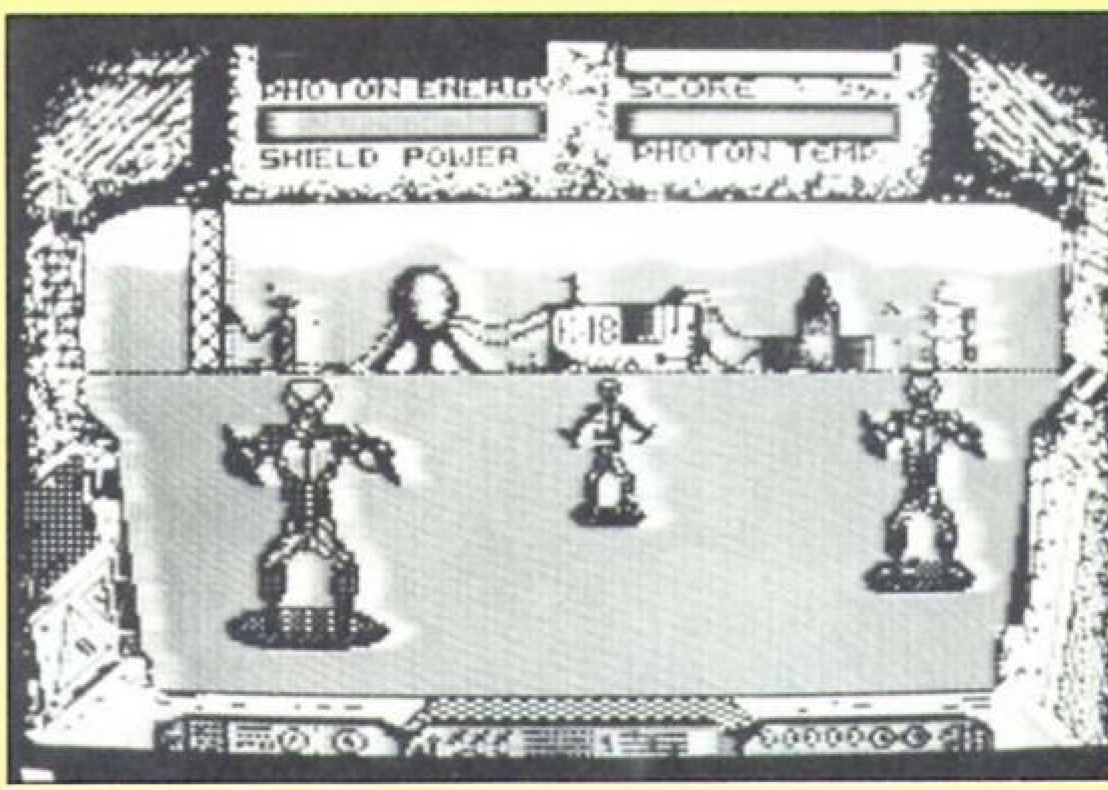
Just put the names of the various victorious superheroes against the appropriate letter on the entry coupon.

The competition is open to all ZX readers except employees of Argus Specialist Publications, Chase Web and Martech. The editor's decision is final and no correspondence can be entered into.

Send your entries to Cosmic Shock Absorber Competition, ZX Computing Monthly, No 1 Golden Square, London W1R 3AB.

Please remember to put your answers on your entry envelope.

The closing date for entries is April 10th.



Cosmic Shock Absorber Competition

The vanquishing superheroes are:

a)

b)

c)

Name ....

Address ....

....

Send your entries to Cosmic Shock Absorber Competition, ZX Computing, No 1 Golden Square, London W1R 3AB.

GAMES

KONAMI'S GOLF
Imagine
£7.95

Konami's Golf is played over a course at the 'KCC' — the Konami Country Club. This rather grand title hides the fact that all we have here are nine holes and a competent but uninspired golf simulation.

All the basic features we've now come to expect are here, options for one player and two players (stroke and matchplay), choice of clubs from a bag of 13, an animated golfer whacking the ball off the tee and powermeter to adjust the length of shot.

The display design attempts to cram everything onto a single screen. The overall view of the fairway on which you choose the direction of the ball could just as well have been put on a separate screen, leaving more room for the main golfers eye view screen display.

There are few factors to take into account when setting up your shot. You can choose to hook, slice or hit straight and the only external influence is wind speed and direction, which in practice, seemed to have negligible effect on shots. When close to the green the display shifts to an overhead view and factors such as slope and speed of the green come into play. These are fairly easy to compensate for and don't really add to the difficulty.

Konami's Golf is only worth shelling out money for is you are desperate to get a Golf simulation into your software collection but with only nine holes, its appeal may fall off very quickly.





MASTER GAMES
US Gold/Ultimate
£9.99

US Gold have just released two compilation tapes of old Ultimate games, both entitled Master Games (and with only the track listing allowing you to tell the two apart). The first of the two contains the first four games that Ultimate ever produced, which were themselves amongst the first proper arcade games ever produced for the Spectrum (and if memory serves, all but one of them load in under 16K as they were produced in the days when the 16K Spectrum was still going strong).

Ultimate's very first game was Jetpac, a simple but highly addictive shoot 'em up in which you have to collect fuel pods and treasure on a number of planets, all of which are swarming with deadly aliens. At the time, Jetpac was head and shoulders above any other game for the new Spectrum machine, with smooth, fast animation and good sound effects. I showed Jetpac to someone who didn't see it when it first came out and their reaction was that it looked a bit dated but was still fun to play.

The same is true of Lunar Jetman, the sequel to Jetpac. This was Ultimate's first 48K game and is a bit more complicated, though it's still essentially a shoot 'em up. This time around the Jet Man has a moon buggy to control, as well as a number of objects to collect and aliens to blast. But while the two Jet Man games have remained enjoyable over the years, the other two titles on the tape have dated rather badly.

PSSST is a variation on the shoot 'em up style in which you have to spray worms with a number of spray cans, while Trans Am was an early driving game. Neither of these games really stand up any more, being about the same standard as the average budget game.

So, whilst the two Jet Man titles are a tribute to the quality of early Ultimate games, this compilation as a whole doesn't really stand out when compared to some of the others available. On the other hand, the second Ultimate Compilation (reviewed elsewhere in this issue) is worth a closer look.





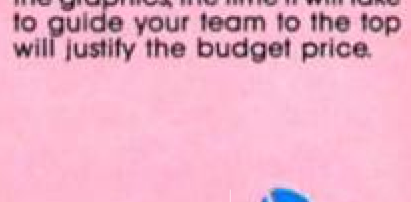
LEAGUE CHALLENGE
Atlantis Software
£1.99

This game falls into the "so bad it's good category". It's a football manager type game, so if the lure of league tables and team selection problems holds no fascination, read no further.

If you've stayed on you can be initiated into the mysteries of a strange new game — "Zombie Laserball". When League Challenge attempts to capture the big match atmosphere we are treated to some very odd graphics indeed. Hulking shapes lumber across the pitch aimlessly like zombies but just when you think they are totally lifeless a laser beam darts out of one of the zombies boots and tries to zap the undead goalkeeper. He misses and it's a goal. These bizarre graphics decide your league and cup success.

You start off in the murky depths of Division 4 obscurity and by buying players, training hard and selecting a strong team, you hopefully rise to glory in the First Division.

Your form can be decidedly quirky, surprise defeats followed by a run of victories after your players suddenly gains lots of stamina and fitness points for no apparent reason. League Challenge will certainly win no awards but if you can excuse the graphics, the time it will take to guide your team to the top will justify the budget price.



HENRY'S HOARD
Alternative Software
£1.99

Alternative Software is the latest company to start producing budget software, and so far their releases for the Spectrum have been passable versions of pretty old arcade games. Henry's Hoard, possibly the best of their first batch of games is a return to that old favourite, the platform game (how long is it since you've played one of those?).

It's a 50 screen game in which you have to send Henry (a small sprite with wiggly legs) off in search of items of treasure scattered around the various rooms. As is the way with these games there are lots of little spikey objects, stairways, moving walkways and deadly monsters to avoid, all by resorting to the fiendish strategy of jumping left and right. As Manic Miner clones go it's fairly average — the graphics are a little on the small side but it's clear enough what's going on, and some of the obstacles require a bit of thought to get past.

Henry's Hoard isn't going to set the world alight, but I've always had a weak spot for platform games (they're so much more refined than those nasty shoot 'em ups), so it gets a modest thumbs-up.





MASTERS OF THE UNIVERSE
US Gold
£9.95

When US Gold bought the licence for this hit TV series it was undecided which type of game would best suit the heroic battles of He-man. The result is two games. This arcade adventure and a Scott Adams style adventure.

Skeletor, arch-enemy of He-man has found the all-powerful Illearth stone and has overthrown Castle Grayskull. You have to enter the castle, find the stone and defeat Skeletor and free the land.

Orko tried to trap Skeletor in an imprison cell which backfired trapping Orko in one of the castle's stone pillars. You must find him and give the right ingredients that he will use to turn your sword into Atom Smasher!

On screen, He-man is represented by a feeble yellow clad figure that is surrounded by a black oblong that blots out his immediate surroundings. However, you haven't long to admire this amazing breakthrough in Spectrum graphics, you are attacked by one of Skeletor's minions. Before you can say "you don't mess with He-man" you've despatched it with a single blow and added one to your "skulls cracked" total. This total lies below the main screen display along with the state of your body shield, the objects you carry, your sword status and the self explanatory countdown of the moons till doom.

The way to the castle is short but hazardous and lies beyond two spiked pits complete with deadly electrical pulses beaming across them. Well timed leaps will lead you to the key and then the castle but mistime one of the three jumps you must make and your quest comes to an abrupt end.

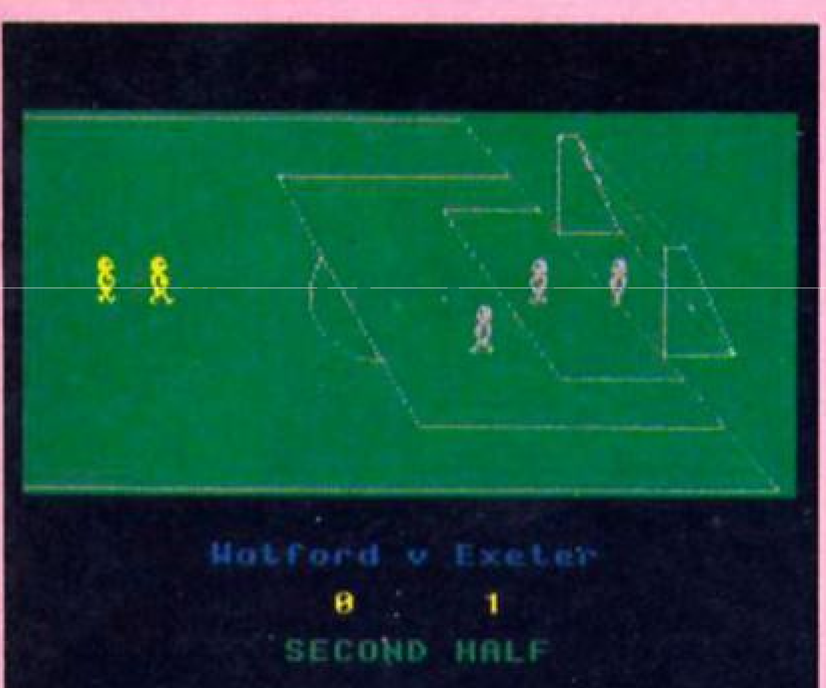
Inside the castle you are met by even more nasties that leap, fly, jump and fire at you but most can be despatched by your fully charged sword of power. This leaves you free to explore the platforms and rooms of the Castle Graystoke until you find Orko with the correct ingredients and defeat Skeletor.

Unfortunately, neither the gameplay or the graphics capture the atmosphere of He-man or the rest of the Masters of the Universe crew and your joystick or keyboard could be controlling anybody or anything (even a pumpkin) while you battle nondescript monsters in yet another completely silent platform screened arcade adventure.

Another wasted licence.







DEFCOM
Quicksilva
£8.95

In the year 2056 it was complete. The Star Wars defence system was at last in position making war on Earth impossible. Weapons were scrapped as the Earth settled down to a life of leisure.

The day before the scheduled white Christmas they came. An alien force that took over the defence satellites that were then turned into a weapon aimed at Earth.

One man could save the world. In Quicksilva's shoot everything that moves game, you are Nick Diamond, Xeno champion and the only man left who can fly an Eagle Class laser firing fighter.

Saving the world isn't going to be as easy as it sounds as you must take out each of the eight satellites that are threatening the world's major cities. They can only be destroyed by a Blaster that is only obtained once you have destroyed a staggering 144 aliens and then you've only got one shot!

Until then you must zap aliens and restrict the destruction by intercepting the satellite bombs with a Dynoray (obtained after only 12 kills).

A menu of options can customise the bland display by adding a score and energy display as well as a VDU that reports of heavy alien activity and a map to plot your course.

The normal Spectrum blips and beeps as the aliens swirl

past can be improved by the sound option if you have a Spectrum 128 or a Fuller or Timedata add-on.

However, most of these options are little more than standard features in other games and can't help a game that falls short of expectations and takes an age to play.





XCEL
Mastertronic
£1.99

XCEL was originally released as a full price game about a year ago and it didn't exactly set the world alight, however as a budget title it's worth a look if



you're into shoot 'em ups. The game has a few bells and whistles attached in the form of some icons, but these don't really shift the emphasis of the game from much other than straightforward alien blasting.

The action takes place on a group of planets which are controlled by Sentinel computers and it's up to you to fly from planet to planet bombing everything in sight in order to return control to the human population. You fly over the scrolling surface of the planets, avoiding trees and other objects in order to blast the computers' control centres, and after each bombing run there's a Galaxians style wave of attacking aliens.

The action isn't as fast and furious as some other shoot 'em ups, and at times the game looks a little dated, but there are about 300 planetary locations to blast your way through so at least there's a fair bit of variety in the game. Mind you, I did find it a bit irritating that each time you get killed you have to go right back to the start of a particular sequence (rather than carrying on from where you got killed), which means that you can find yourself going over the same screens over and over again.

Still, the game is nicely presented and at £1.99 is much better value than when it was first released.



THE HIVE
Firebird
£9.95

The Queen of the hive must be slain and it's your job to do it in Firebird's latest twist to the maze game. The Queen is in the middle of a three dimensional hive that turns and spirals through eight complex and entangled levels.

You pilot a versatile electro-mechanical Grasshopper ship that runs, jumps and crouches its way through the maze.

Just running through the maze is hard enough without the need to map (to make sure you don't run around in circles) and avoid or blast the nasties that are naturally out to get you. These include swarms of electronic insects, giant demonic jailers and spikes, webs and stings that block your path.

Contact with any of these will drain your energy that can be replenished in you have two or more energy pylons that can be found in the maze to replace your original supply. If you have only one pylon then damage is not repaired. Losing them all brings the inevitable end as your energy is quickly drained away.

The screen display shows your view down the tubes and instruments that show your current energy level, score and any objects, pylons, shields and weapons you may have collected.

Lurking underneath the 3D display is yet another maze game that has been complicated by the unpleasant key controls that make steering the Grasshopper a major achievement.

Finally should you want to save your efforts then you only have to copy down a code number that can be entered when a new game is loaded.





AGENT X
Mastertronic
£1.99

This turned out to be a pleasant surprise — another Mastertronic cheapie that compares well with many of the expensive games that were churned out for Christmas.

Agent X loads in several parts, each piece of coding being a different stage in the game, and so the game manages to pack quite a lot into just £1.99's worth of cassette. The game starts with a furious car chase as Agent X sets off in hot pursuit of the kidnapped President. The graphics in this sequence are excellent — a large and nicely detailed 3D representation of a highway with various vehicles travelling in both directions and getting in the way.

The action is a bit chaotic, and getting through this section is more luck that judgement until you get the hang of it, but it's all so fast and furious that it's fun anyway.

The next sections of the game take you into the mad professor's lair, where you have to take on the security men (using an extremely silly-looking flying kick manoeuvre) and then into confrontation with the prof himself as he tries to brain you with anything that comes to hand.

It's all rather silly really, but the graphics are good, the game is nicely presented and the programmer's clearly kept his tongue well into his cheek, making this Mastertronic's best budget game for months.





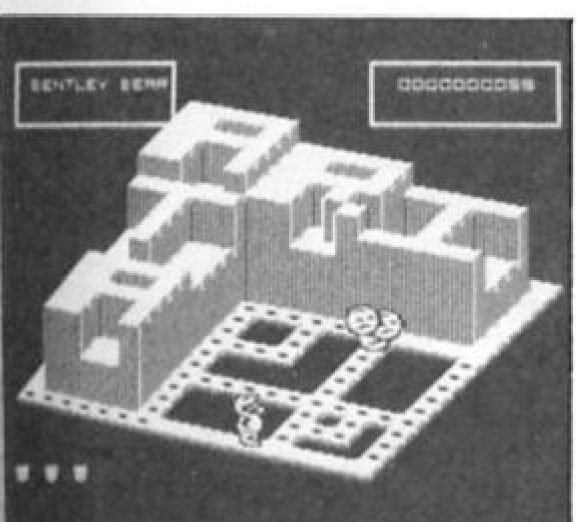
CRYSTAL CASTLES

A software 'legend' from U.S. Gold?

Crystal Castles
U.S. Gold
£8.99

U.S. Gold are releasing this as a 'special limited edition' in an attempt to convince us all that if we don't rush out and buy it straight away we will all have missed our chance to experience this "software legend" (their advertising people certainly know their stuff).

The trouble with legends is that they don't always measure up to their reputations when you eventually get to see them in the flesh — especially when they're being translated from a dedicated arcade machine onto the Spectrum. I'd never seen the arcade original but someone told me that it was a sort of 3D Pac Man. That sounded promising, despite the fact that the main character is



called Bentley Bear, so in went the tape.

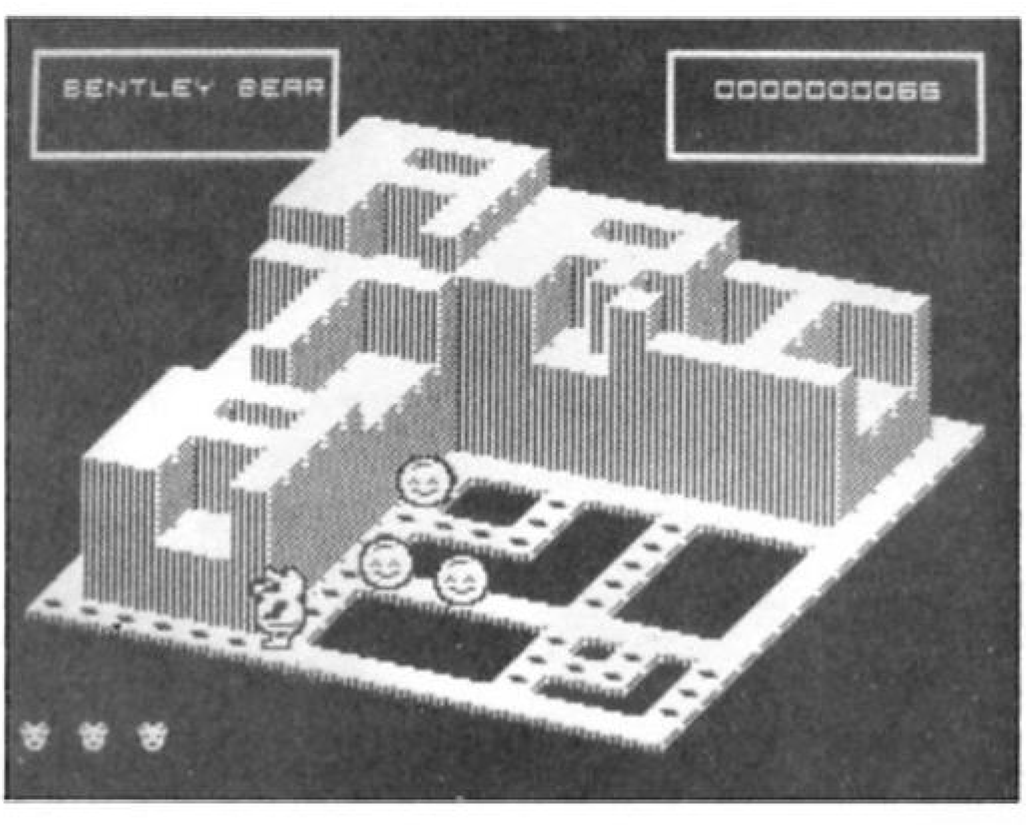
The 3D Pac Man description was fairly accurate in that Bentley's task is to wander around a three-dimensional 'castle' which is made up of structures of ramps and walkways which become increasingly complex on each successive screen, collecting little pills which are dotted along the main pathways just as in Pac Man. As you'd expect there are various types of monsters that home in on Bentley (I hate that name), including things that look like centipedes, trees and ghosts, and also additional items of treasure that can be collected or left where they are in order to block the path of the pursuing monsters.

Of course, as each screen is in three dimensions rather than the original Pac Man's two, the game is a little more complicated. Bentley has the ability to jump over monsters rather than simply moving left/right, forward/backward, and it quite often happens that he will move behind a section of the 'castle' structure. When this happens the program allows you to 'see through' the structure so that you can follow Bentley's movements although you are unable to see the actual path that he is on.

The drawback with using 3D structures is that the graphics are that much more complicated and in order to get everything onto the screen the moving figures and blocks which make up the structures all have to be quite small. This makes the game quite fiddly to control. Most of the paths around the screen are so narrow that it's not at all unusual for Bentley to shoot past one when you actually want him to turn into it. So you have to slowly double back and line him up precisely, before going down the path. By this time, of course, every monster on the screen has homed in on the hapless bear and sent him to bear heaven. As a result of this fiddliness the game's addictiveness suffers considerably. Instead of trying to do an exact conversion the programmers of the Spectrum version would have done well to keep the castle structures a little less complicated (along the lines of Addictive's Kirel, which had similar 3D structures but ones which were drawn larger and more clearly, yet were still complex enough to present a challenge).

By trying to pack a bit too much onto the Spectrum's screen the programmers have made the game too cramped to allow you to build up some speed and get really involved in it.





SHORTCUTS

Ray Elder presents more programming gems from ZX readers.

M/C Break

First up this month is a program from Andrew Vellacott of Essex who got so fed up with his machine code programs getting stuck in infinite loops during development that he wrote this short routine which makes use of the Spectrum interrupt mode two to scan for the BREAK key being pressed.

Two minor points to note: First you will have to reinitialise the routine each time after BREAK is pressed with RANDOMIZE USR 32377, and secondly, machine code or BASIC loader programs must not occupy the addresses from 32348 to 32375.

A useful aid for machine code programmers.

M/C Break

```
   1 REM M/C Break
  1Ø DATA 255,243,245,62,127,219
,254,2Ø3,71,32,15,62,254,219,254
,2Ø3,71,32,7,241,237,66,251,195,
3,19,241,251,2Ø1,62,4Ø,237,71,23
7,94,2Ø1
  2Ø CLEAR 32347
  3Ø RESTORE: FOR i=32348 TO 323
75: READ a: POKE i,a: NEXT i
```

Font

```
   1 REM Font
  1Ø LET MC=6ØØØØ: LET CHR=55ØØØ
  2Ø FOR f=MC TO MC+26: READ a:
POKE f,a: NEXT f
  3Ø DATA 17,Ø,2ØØ,237,83,54,92,
33,Ø,6Ø,126,18,254,6Ø,32,+3,62,1
26,18,19,35,124,254,64,32,-16,2Ø
1
  4Ø RANDOMIZE CHR: POKE MC+1,PE
EK 2367Ø: POKE MC+2,PEEK 23671:
LET L=USR MC
```

Font

R. Glavas has been a regular contributor from the start, his offering this month is a variation on a frequently used theme, that of a redesigned character set. What we liked about this set was that it was changed subtly, giving a more square effect.

Simply enter the program and RUN it, the character set resides at address 55000.

Scroll Relocate

One of the things I like about this column is the way readers often send in and share modifications of previous programs. R. Dorsett of Middlesex sent us a short cut which solves a problem he found with Leif Kvernhusvik's Scroll program in the DEC 86 issue.

The original program is located in the printer buffer, not good news for printer owners, so Mr. Dorsett's program will relocate it to anywhere in memory that you may desire.

First (providing you have a M/C copy of the program on tape) CLEAR ramtop to the new start address−1. As an example we will imagine that we are going to move the routine to address 60000, therefore CLEAR 59999.

Now LOAD "SCROLL m/c" CODE newaddress (EG. LOAD "SCROLL" CODE 60000).

Type in the RELOCATE program and RUN it.

Finally SAVE "SCROLL" CODE newaddress, 159. (EG. SAVE "SCROLL" CODE 60000,159).

Don't forget to change the line in your program which defines the function to call the new address. (EG. DEF FN x(a$,1,w,a)=USR newaddress.

Some alterations to the way in which the program works can also be made: POKE start+125,201 will return the program to BASIC if a key is pressed or when the scrolling is finished. POKE start+118,0:POKE start+125,201 prevents the program from being interrupted and returns to basic when finished. POKE start+118,192:POKE start+125,195 and the program will only return to basic when a key is pressed.

Scroll Relocate

```
9899 REM Scroll Relocate
99ØØ INPUT "Start Address ";s
991Ø FOR n=1 TO 7: READ a,b: LET
 a=a+s: LET b=b+s
992Ø POKE a,b-256*INT (b/256): P
OKE a+1,INT (b/256)
9925 NEXT n
993Ø DATA 1Ø,12Ø,27,144,3Ø,159,5
9,161,7Ø,159,73,161,126,Ø
```

Mirror

S. Neill of Angus, Scotland, provides us with a means of turning the screen image in a mere 19 bytes of machine code. Enter the program MIRROR 1 and RUN it. Type NEW and enter MIRROR 2 which will demonstrate the effect. Note that the Machine code is located at address 30000 and can be saved by SAVE "MIRROR" CODE 30000,19.

Specific sections of the screeen can be mirrored by POKING the following values to the addresses shown. x=start address

Whole screen — POKE x+2,64:POKE x+15,88
Top third — POKE x+2,64;POKE x+15,72
Middle third — POKE x+2,72:POKE x+15,80
Bottom third — POKE x+2,80:POKE x+15,88
Top two thirds — POKE x+2,64:POKE x+15,80
Bottom two thirds — POKE x+2,72:POKE x+15,88

Mirror 1

```
   1 REM Mirror 1
  1Ø CLEAR 29999
  2Ø FOR n=3ØØØØ TO 3ØØ18
  3Ø READ a: POKE n,a: NEXT n
  6Ø DATA 33,Ø,64,78,6,Ø,2Ø3,17,
31,16,251,119,35,124,254,88,32,2
41,2Ø1
```

Mirror 2

```
   1 REM Mirror 2
  1Ø LET a=12Ø: LET b=9Ø: LET r=
8Ø
  3Ø FOR n=Ø TO r
  4Ø LET z=INT (.5+SQR (r*r-n*n)
*2)
  5Ø LET x=b+z/2
  6Ø PLOT a+n,x: DRAW Ø,-z
  7Ø PLOT a-n,x: DRAW Ø,-z
  8Ø NEXT n: RANDOMIZE USR 3ØØØ
  85 STOP
  9Ø SAVE "DEMO" LINE 9ØØØ: SAVE
 "Mirror"CODE 3ØØØØ,19
  92 STOP
9ØØØ CLEAR 29999: LOAD "" CODE :
RUN
```

Perspective

Mark Batts who resides in Watford sent us this little program which could enhance title screens or messages.

Perspective

```
   5 REM Prespective
  1Ø LET X=Ø: LET Y=Ø: LET E=2:
LET F=16Ø
  15 INPUT "Type a word please n
o more than 6 letters ";A$
  2Ø IF LEN A$>6 THEN GO TO 15
  4Ø PRINT AT Ø,Ø;A$
  5Ø FOR S=Ø TO LEN A$*8
  6Ø FOR T=175 TO 168 STEP -1
  7Ø IF POINT (S,T)=1 THEN GO SU
B 11Ø
  8Ø LET Y=Y+E: NEXT T: LET Y=Ø:
LET X=X+E: NEXT S
 1ØØ LET F=9Ø: LET E=2: LET X=Ø:
 GO TO 5Ø
 11Ø PLOT X,F-(Y+E): DRAW Ø,E
 12Ø PLOT X,F-Y: DRAW E,Ø
 13Ø PLOT X+E,F-Y: DRAW Ø,-E
 14Ø PLOT X+E,F-(Y+E): DRAW -E,Ø
 145 LET E=E+.Ø5
 16Ø RETURN
```

Reflections

The above mentioned Mr. Batts supplied us with a program in a previous issue called reflections which reader Stephen Carr has modified and improved. Another attractive way of presenting short text messages.

Reflections

```
   1 REM Reflections
  1Ø LET Y=Ø
  2Ø BORDER Ø
  3Ø PAPER Ø
  4Ø INK 7
  5Ø CLS
  6Ø PRINT "Type in a word up to
ten """"letters long-"
  7Ø INPUT A$
  8Ø CLS
  9Ø IF LEN A$>1Ø THEN GO TO 6Ø
 1ØØ PRINT INK Ø;AT Ø,Ø;A$
 11Ø PLOT Ø,75
 12Ø DRAW 255,Ø
 13Ø LET D=LEN A$*8
 14Ø LET X=127-(D*3/2)
 15Ø FOR S=Ø TO D*8
 16Ø FOR T=168 TO 175
 17Ø IF POINT (S,T)=1 THEN GO SU
B 24Ø
 18Ø LET Y=Y+5
 19Ø NEXT T
 2ØØ LET Y=Ø
 21Ø LET X=X+3
 22Ø NEXT S
 23Ø STOP
 24Ø FOR L=3 TO 1 STEP -1
 25Ø CIRCLE X,75+Y,L
 26Ø CIRCLE X,75-Y,L
 27Ø NEXT L
 28Ø RETURN
```

Versatile Input

Being able to position an input prompt and then accept characters and display them immediately after the prompt ANYWHERE on screen and not just at the bottom of the screen can give your program a professional touch.

Andrew Smith of Cleveland provides us with a means of doing just that.

The subroutine is located at lines 9000 plus but can be renumbered to suit your own needs. To use it set the screen position and provide a prompt (ensure you end with a semicolon else the input will be on the wrong line) and GO SUB the routine, for example:
100 PRINT AT 5,10;"ENTER YOUR NAME ";:GO SUB 9000

The resultant entry will be returned in variable i$. The variable z$ is used by the routine to act as a temporary input string, the DELETE key will delete a character and the ENTER key terminates the input. A nice bit of well error trapped BASIC.

And so once again we reach the end of another collection of small, but beautifully formed, programs. All that remains is to nominate our Short Cut of the month. The prize goes to Mr. Neill for his Mirror program.

Input

```
9ØØØ REM .....INPUT
9Ø3Ø LET i$=""
9Ø4Ø PAUSE 2Ø: PRINT OVER 1;"_";
CHR$ 8;
9Ø5Ø LET z$=INKEY$: IF z$="" THE
N GO TO 9Ø4Ø
9Ø6Ø BEEP .ØØ5,3Ø
9Ø7Ø IF z$=CHR$ 13 THEN PRINT "
";CHR$ 8;: RETURN
9Ø8Ø IF z$=CHR$ 12 AND LEN i$>Ø
THEN LET i$=i$( TO LEN i$-1): PR
INT " ";CHR$ 8;CHR$ 8;"_";CHR$ 8
;: GO TO 912Ø
9Ø9Ø IF z$=CHR$ 12 OR z$<" " OR
z$>"z" THEN GO TO 912Ø
911Ø LET i$=i$+z$: PRINT z$;"_";
CHR$ 8
912Ø IF INKEY$=z$ THEN PAUSE 5:
PRINT OVER 1;"_";CHR$ 8;: GO TO
912Ø
913Ø GO TO 9Ø4Ø
```

STALLONE COBRA

Cobra
Ocean
£7.95

Sylvester Stallone converted into fun cartoon character . . . Strange but true as Ocean unveil an off-beat spin-off



This is a movie spin off game that stands up in its own right, perhaps because it bears so little similarity to the original film.

Cobra, a vigilante cop movie, with Stallone as the star, was a failure by Rambo standards although it contained the by now standard quota of mindless violence. Ocean have made no attempt to make a carbon copy game of the film and in fact Cobra is a tongue in cheek send up with a ludicrous plot, absurd characters and cartoon logic.

The cop, "Cobretti", is a musclebound chunky sprite who is initially armed only with a devastating headbutt which if he connects send his enemies winging off screen in the direction they came on. As for these legions of psychopaths, they are made up of obese wrestlers, old ladies touting lethal bazookas, knife throwing assassins and the like. This is not to mention the turbo charged prams that ram Cobretti and sap his energy.

If all this sounds a bit wacky, wait for the plot. Okay . . . Cobretti has to rescue top fashion model Ingrid Knutson from the clutches of the evil Night Slasher. All right so far but in order to do this he's got to find weapons which are hidden inside beefburgers at certain points in the three playing areas (city, country and a factory). Quite why a knife, pistol, and laser sighted machine gun are concealed in this way is a mystery. Anyway shaking off the mince from the weapons, Cobretti must rescue Ingrid and completely decimate the attackers before progressing to the next section. The weapons however have a limited life span and tend to disappear just when you need them most.

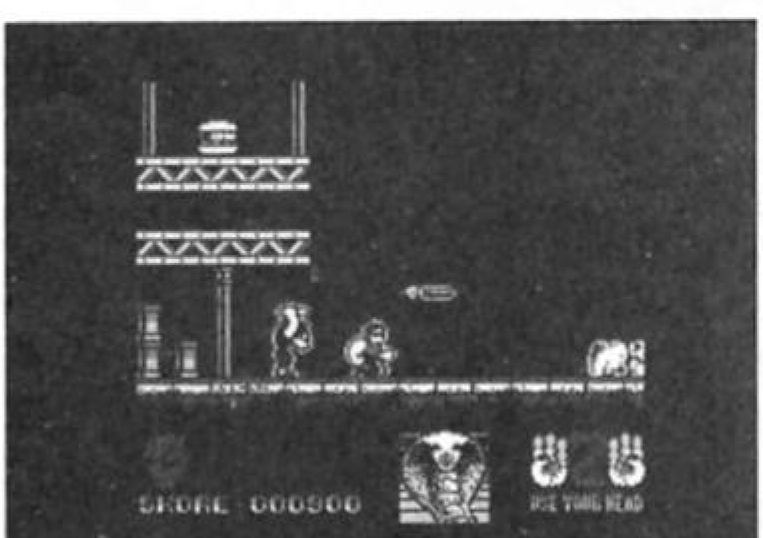
Cobra is a very fast, very well animated game that's hard to master at first simply because you are confronted by so many bizarre comic hatchet men.

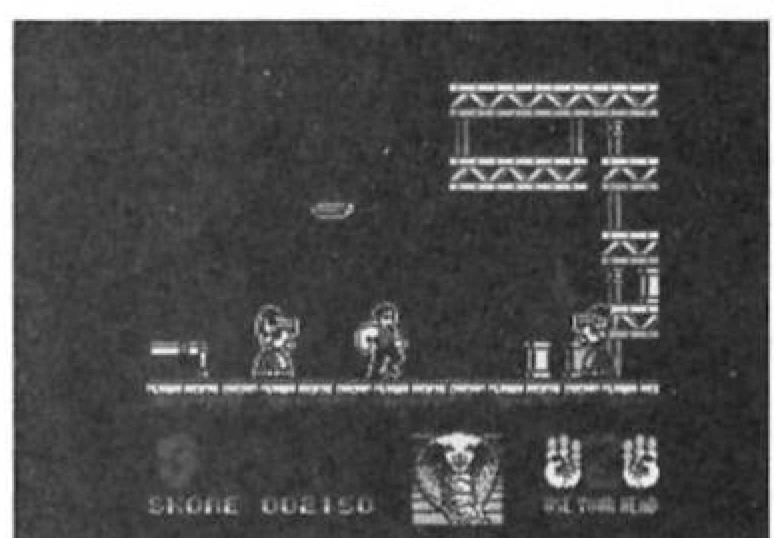
You are given a miniscule three lives to start out with and additional lives can be obtained at 10,000 points and further up the high score table.

The game, which can be played with keyboard or joystick, is really enjoyable and arcade fans who are looking for something a little bit different will not be disappointed. Unfortunately the image created by the film is misleading. Stallone on the inlay cover with his laser sighted machine gun and the hard man phraseology "Crime is a disease. He's the cure", look a bit silly once you've got the Stallone figure scampering up the platforms after the beefburger.

It may well have been the intention to send the whole thing up and that's fine but those who like their destructive blasting untainted by any form of fun had better stick to the film.









It's time to test your space pilot prowess at the Academy.

Academy
CRL
£8.95

After an unfortunate incident on 61 Cygnus when a rookie pilot messed up his approach and half the planet when he tried to dock with its central reactor the Gal-corp set up the Academy. Its aim was to train rookie pilots to produce an elite corps of pilots and prevent such disasters while providing a new challenge and sequel to CRL's excellent Tau Ceti.

To graduate from the Academy a cadet must complete 20 missions and achieve an average rating of over 90%! The missions are grouped into five levels of four missions that are stored on a separate data tape. Most players won't need this tape as they'll still be struggling with the first four that are loaded in with the master program.

A main menu allows you to register as an Academy cadet, view and select the missions and the skimmers to complete them.

The first four missions range from the less than subtle "If It Moves . . ." annihilation run to the search through a minefield for base mission called "Softly, Softly". "Meltdown" is a race against time to close down a critical reactor on a polar world, finally "Red Dawn" a Tau Ceti style mission in which you must destroy robot factories in all quadrants connected by jump pads.

Whichever mission you choose (and you have to tackle them all sooner or later) you'll be attacked by robot droids of both familiar and new designs. Four types of laser firing hunters lead the assault supported by mines, super mines, droids, hoppers, guardians, trackers and aptly named kamikazes. Therefore it's important that your Gal-corp skimmer is armed and shielded for the mission. Three skimmers are ready for take off, each offering a different range of equipment from the GCS Wilson complete with jump unit to the weapons and shields tank known as the GCS Lenin.

If neither of these or the balanced GCS Lincoln suit your needs or tactics then you can design your own as long as it remains within the weight and cost limits set down by the Academy.

Your basic skimmer consists of low, medium or high powered laser, main drive, shield and steering units which you can then arm with missiles, anti-missiles and delay bombs then add whatever anti-missiles, jump units, scanners, compasses and trackers that you need and can afford. Finally you can customise the skimmer view screen and save your amazing design to tape for further missions.

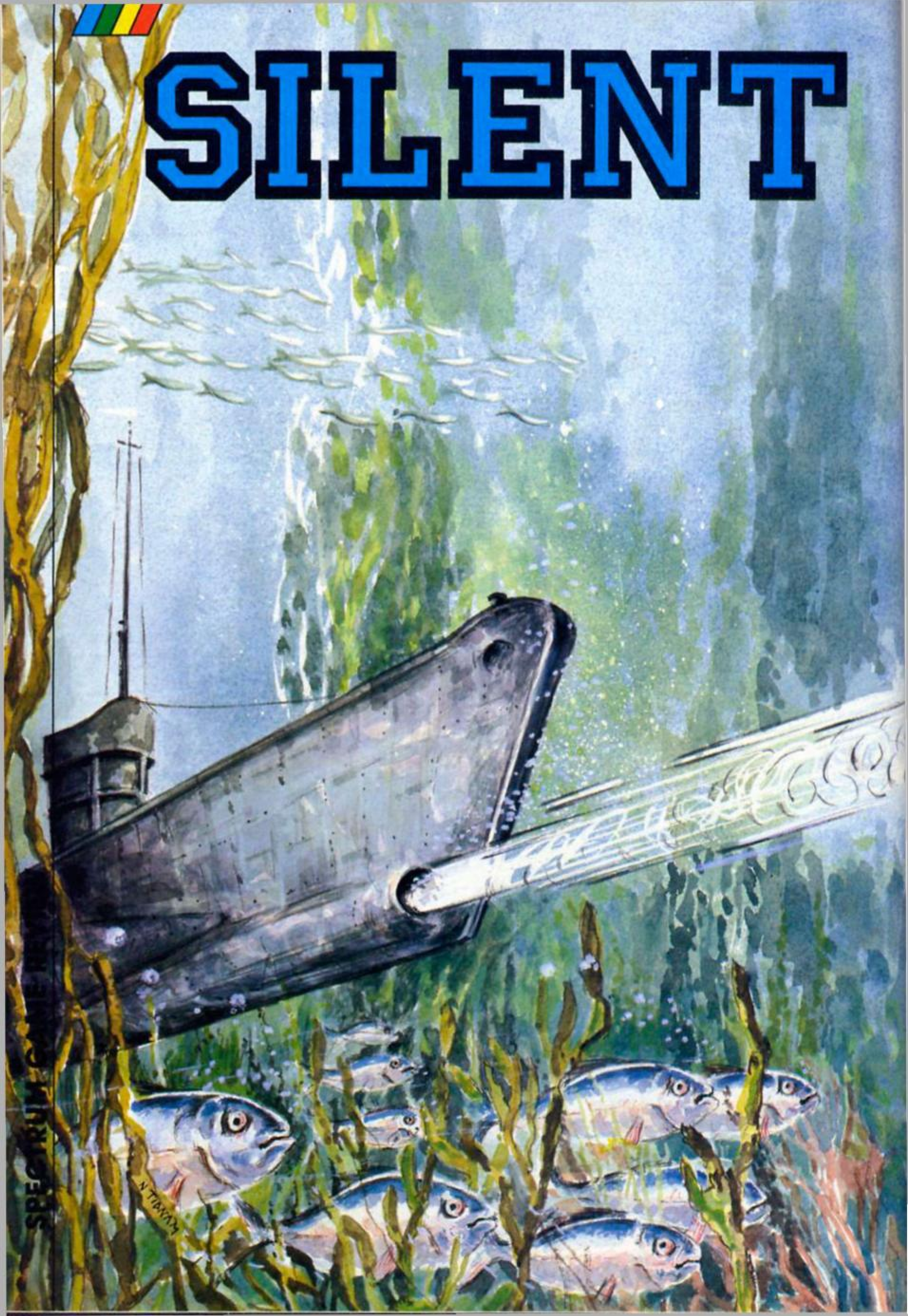
Once the newly constructed GCS Rambo had wiped out the robot factories and mines that stood in it's way I was ready for the next set of missions but still a long way to go before graduation. Ahead lay the dubious delights of such encounters as ". . . at the OK Corrall", "Don't Panic", "Hide and Seek", and "Mission Improbable".

Finally as a bonus the gamepack also includes a star map program so you can find your favourite stars and constellations. A game featuring 20 missions each as challenging as the original game doesn't need a bonus program.

MONSTER HITS






SILENT
SPECTRUM

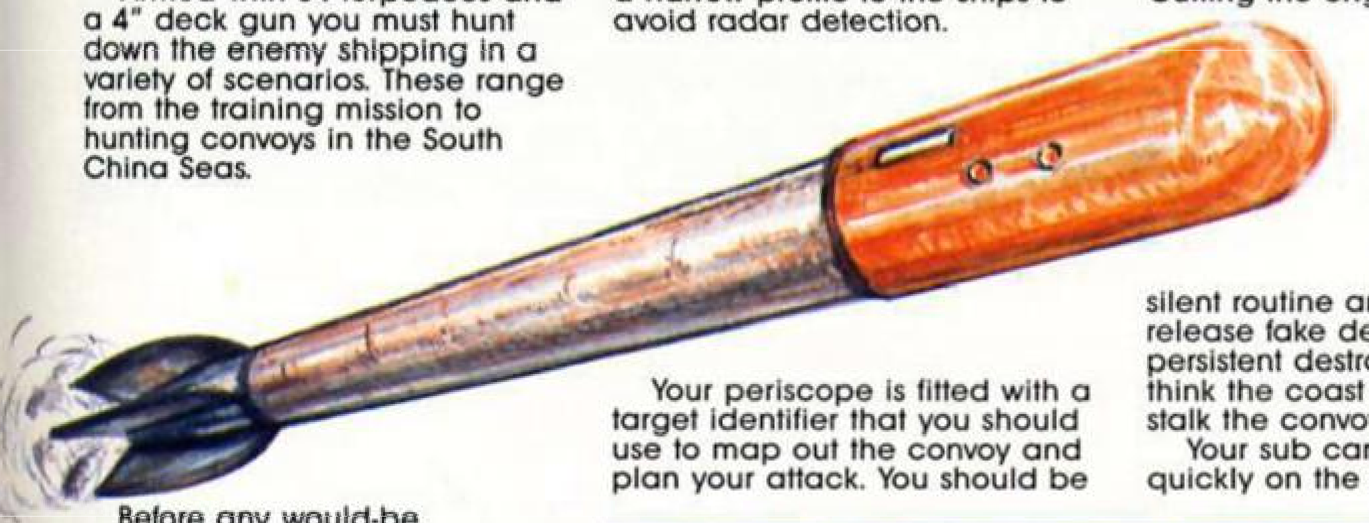
SERVICE

Action above and below the waves in US Gold's sub simulation.

Silent Service
US Gold (Microprose)
£9.95

Commanding a US submarine during the Second World War, your mission is to sink as many Japanese ships as possible while avoiding the attention of enemy destroyers.

Armed with 34 torpedoes and a 4" deck gun you must hunt down the enemy shipping in a variety of scenarios. These range from the training mission to hunting convoys in the South China Seas.



Before any would-be Submarine Commander can help America win the war (again!) he must first learn how to control the ship. This means frantic use of the keyboard (despite the frequent mentions to a joystick in the instructions). In all, 33 key controls must be mastered before you can terrify Japanese shipping,and so practice is essential using the torpedo/gun training mission. The object is simply to destroy the four old cargo ships that are anchored in position to make things easier!

Pressing Caps Shift and one of the number keys moves you between the battle stations on your submarine, that include a navigation map, bridge and deck gun, periscope for aiming and firing torpedoes, instrument and gauge displays, damage reports and a captain's log that reports your "kills".

As you move between these stations you must plot your course control, your engine speeds and submarine depths as well as planning your attack run.

Convoy Attack

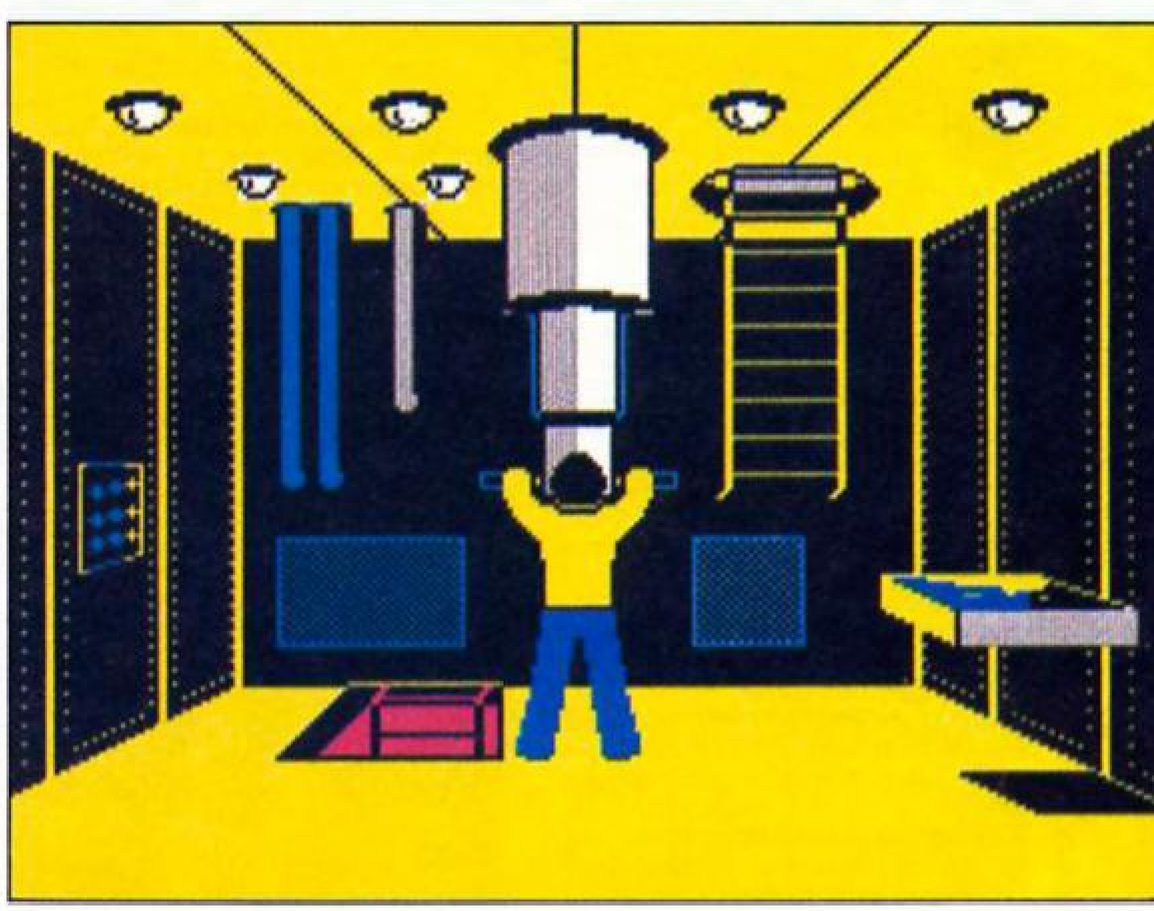
How you attack the convoy depends on the conditions (day/night) and how well it's protected. An unprotected convoy is easy and you can steam up to it at full speed and blast anything that moves. Although you should attack its centre, so you can use both sets of torpedo tubes. Be ready for a chase though as the convoy will scatter.

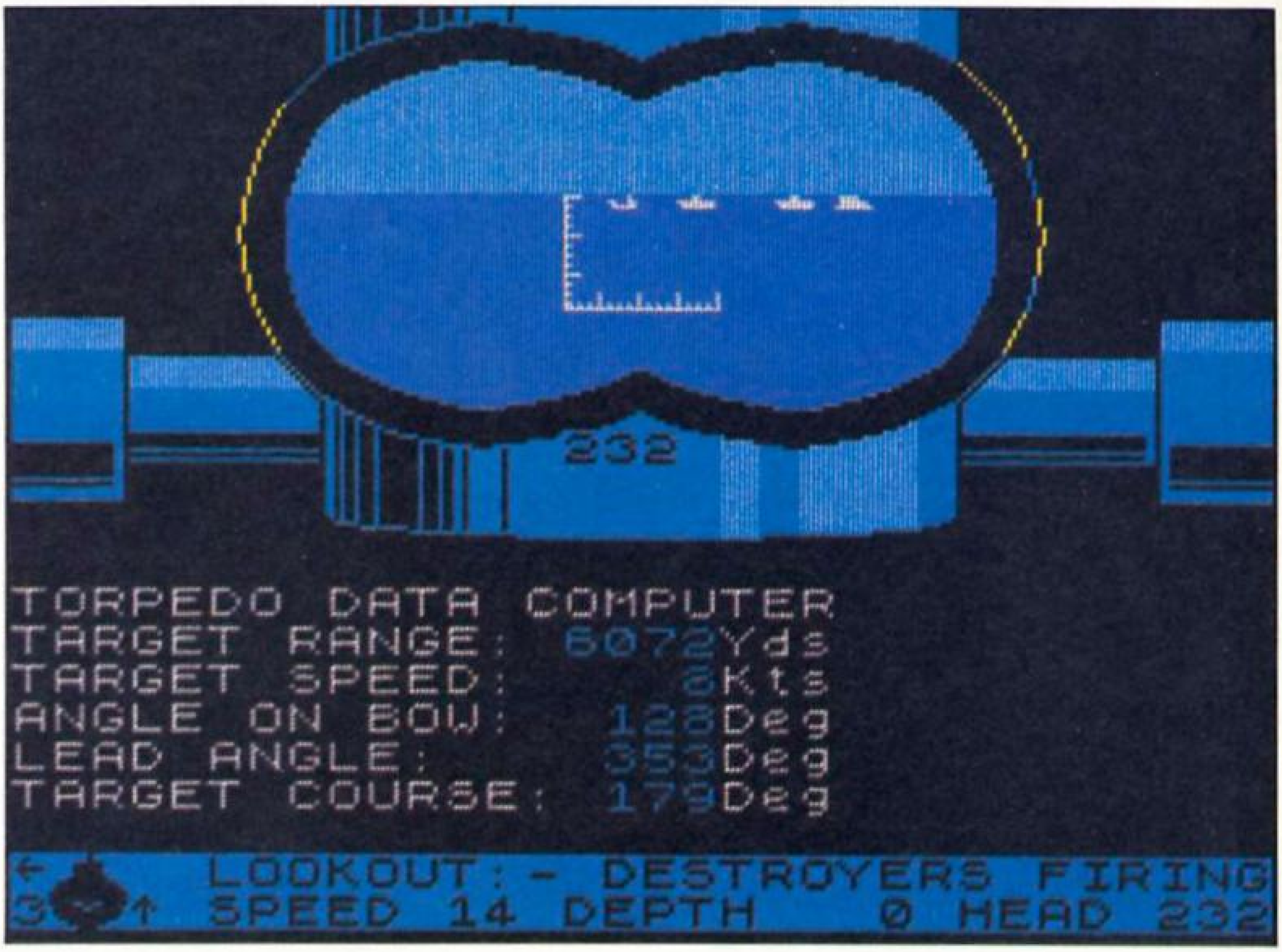
A protected convoy is more of a challenge particularly if there's more than one destroyer. Now you must stalk the convoy at slower speeds while keeping a narrow profile to the ships to avoid radar detection.

Your periscope is fitted with a target identifier that you should use to map out the convoy and plan your attack. You should be able to plot your attack run, so as the destroyer is stuck helplessly on the wrong side of the convoy, giving you a chance to strike and get away. Your periscope also gives you speed, course and distance of target so you can make every shot count.

If there's more than one destroyer and you can't find a gap then you'll have to take one out. This isn't going to be easy as you'll only have one shot before the destroyer locks onto you with it's deadly depth charges. Then it's DIVE, DIVE, DIVE, as you head for the safety of the depths. Cutting the engine will start the silent routine and you can even release fake debris to deceive a persistent destroyer. When you think the coast is clear, begin to stalk the convoy again.

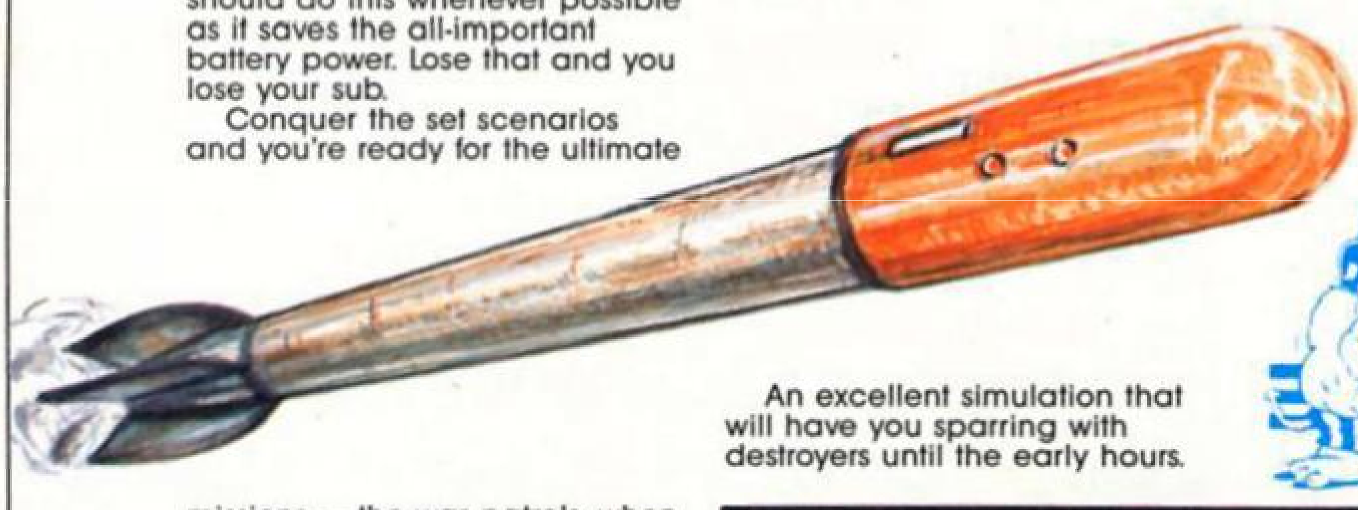
Your sub can travel twice as quickly on the surface and you





should do this whenever possible as it saves the all-important battery power. Lose that and you lose your sub.

Conquer the set scenarios and you're ready for the ultimate missions — the war patrols, when finding the enemy, is as difficult as sinking them. Obviously, lengthy voyages can be time consuming so you can speed up the action to two, three or four times the normal rate until you're ready for battle.

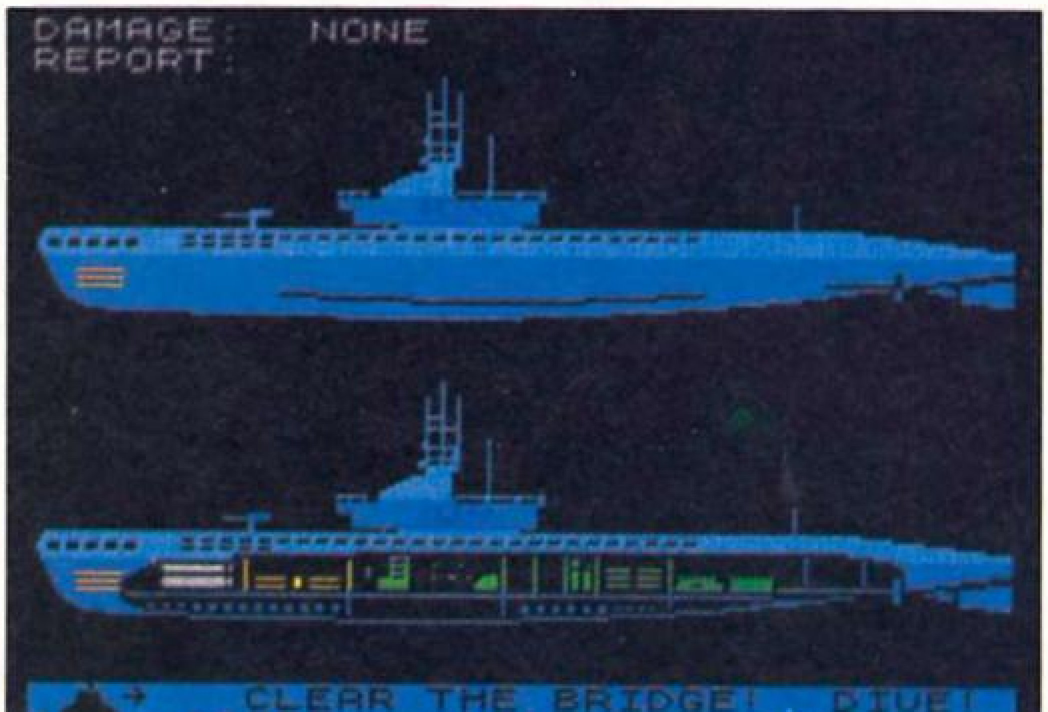


Finally you can make things almost impossible for yourself by adding in skill and reality levels which include limited visibility, dud torpedoes, zig-zagging convoys, no on-ship repairs and the worrying expert destroyers.

The game cassette is accompanied by two instruction sheets that explain almost everything from the control keys to convoy routes, tactics and a submarine commanders' spotters guide to Japanese shipping. When these sheets are folded there's barely room for the game cassette in the box!

MONSTER HIT

An excellent simulation that will have you sparring with destroyers until the early hours.



Brian Beckett with news of Digital Precision's new Turbocharge Compiler.

"Compiler Wars" does not star Harrison Ford, neither does it feature overly cute robots or assorted other humanoids, aliens, Jedi or whatever. Nor is it a computer game in the strict sense — although it certainly is in a wider one — but it promises to be just as amusing and probably just as long lasting. A while ago, Liberation Software released *QLiberator*, a superBASIC compiler to compete with Digital Precision's long-standing and very well received *Supercharge. Both sell at £60 and although QLiberator lacks some of supercharge's finer features and is generally slow it is a good product which attracted a lot of interest.*

In the meantime, Digital was developing Turbocharge. This son-of-*Supercharge*, mega-compiler is due for release at any time. Actually it's been due for release at any time for a while, and some reviewers have taken to needling the company a bit. According to Digital, *Turbo* will be significantly faster than *Supercharge* and have numerous additional features and refinements, designed to justify the package's sub-title of the "encyclocompiler". *Turbo* will sell for £84.95 which includes *Turbo Toolkit*, a package of over 60 new commands designed to extend and otherwise supplement Superbasic. *Turbo Toolkit* is available now (for £24.95) but there doesn't seem much point in buying it yet if you're at all interested in getting the Turbo compiler when it's ready. Digital has sent me the manual so *Turbo* must be about due, since they never bother to write me unless they're about to release something (I didn't even get a Christmas card) for me to tell you about.

In the meantime, the war of words is heating up. Digital has apparently been annoying Liberation Software by constantly referring to *QLiberator* as a "pseudo-compiler". Digital has responded with a denial (possibly at a lawyer's request) stating that the term is simply a technical one referring to the "pseudo code" interpreted by *QLiberator* compared to the "true 68088 code" produced by "true" compilers like (surprise) *Supercharge* and the up-coming compiler-to-end-all-compilers, *Turbo*. The funny bit was that this gentlemanly qualifier was printed in a typeface and layout so small that it practically takes a high resolution microscope to read it at all. Beastly remarks about the competition emerge regularly from both sides and are likely to increase as time goes by. Although it might prove a bit confusing to all those potential customers out there, it promises to be loads of fun for reviewers bored with standard PR releases.

Obviously we'll be looking at *Turbo* when it does come out but, for a sneak preview, here are some of the (many) features Digital is promising. Normally I'd hesitate a bit before writing about a product I haven't seen yet but, as Digital regularly releases high quality packages for the QL, I have little doubt that the product will live up to its promises. First of all, *Turbo* will be multi-tasking (as is *QLiberator*) and — a real favour to the user — will lack that internal lenslok security system found on *Supercharge*. It promises to be much faster than *Supercharge* and is said to be far more user-friendly with an extensive menu and built-in flexibility of use. It will adjust for many structural faults in the programs under compilation and list corrections made. If you're working with an unexpanded QL, lengthy programs are compiled in modules as would be expected but there are a number of features designed to minimise the drawbacks and difficulties of the task. *Turbo* tasks will be able to both read and write into other *Turbo* tasks. Error warnings will be reported by line,, statement and position. This is just the bare surface of what's promised but gives a hint of what's coming over the horizon in the (hopefully) very near future.

I have, however, seen the manual which, although nobody is going to go out and spend £85 for a bunch of A4 sheets, is an impressive feature in its own right. The whole thing is some 350 pages (including the *Turbo Toolkit* manual) and is very well written in a readable and humorous style. There are some cartoons and a glossary (which includes a definition of "reviewer" that makes up for in accuracy what it lacks in basic human kindness) clearly designed to amuse as well as inform. The point is that Digital obviously wants to sell its product to as wide a circle as possible and has gone to a considerable effort to produce suporting documentation that will enable non-programmers (or those less familiar with all the technical ins and outs and various specialist lingo) to get the best out of it. There are detailed explanations of some of the background maths and supporting functions which underlie what is going on and much of the manual is clearly aimed at those not intimately aquainted with the QL. Professional software packaged for all users is a virtue that commends itself and Digital (which always takes this approach) deserves full marks for doing this as a matter of course. The *Turbo* manual excels in this respect.

If you want another QL to play around with the networking facility (which has sort of died a death) or simply because you haven't yet got around to buying one, I've seen them for sale at £100 on the High Street. Push it a bit and you can probably get one for a fair bit less if the retailer has gotten sick of the thing on his shelves (and I suspect most have) and wants to get rid of it. More importantly, it ought to be time to go shopping for a bargain-basement memory expansion and/or disc interface. Although most (but by no means all) of the QL software is designed for an unexpanded machine, just about all the good stuff works a lot better or easier with extra RAM. There's a lot of competing memory expansions on the market and most of the companies are probably beginning to flirt with the idea of getting rid of any surplus stock as the market starts to decline.

At the last couple of Microfairs, I noticed quite a few Medic disc interface/extra RAM modules in service. Medic made an excellent package but sadly ran into difficulties but it looks like some of the original stock of modules has surfaced. If you get a chance to get one cheaply (and I wouldn't pay much for one now) get it because it's probably the best designed QL expansion package I've seen. You will have to find someone to maintain or repair it but, if you can get it cheap enough, it's well worth it as long as you make sure it works to begin with.

QL COLUMN

SPECWORD 48/128

The third and final part of Stuart Nicholls' menu driven word processor including the operating manual.

This month we put the finishing touches to Specword with the final chunk of machine code followed by full instructions on operating the utility.

If you have followed the program through you should by now have saved a BASIC listing, and a machine code listing (HEXDUMP LIST TWO).

Full instructions on entering and Saving Specword were printed in Part 1 (ZX January) but to recap — Using the HEXLOADER used to enter LIST TWO last month enter this month's machine code as HEXDUMP LIST 3.

Save this block of code as "code2" CODE 44800,2519.

When you are happy that all has been saved and verified then reset your Spectrum and enter CLEAR 32767 as a direct command.

Load the BASIC PROGRAM.
Load the Code code 1.
Load the CODE code 2.
Finally SAVE the complete program using "SAVE "WP48/+/128" LINE 9800:SAVE "specode 128" CODE 32768,14600.

Should any of the functions not work then the machine code can be checked using the HEXLOADER/HEXDUMP dual program. The HEXDUMP option will print out to the screen or printer any block of memory in the same form as the HEXDUMP LISTS 2 and 3. (The easiest way to check the code is to compare checksum bytes).

If all has gone well you are now ready to use Specword.

Specword 48/128 Operating Manual

Specword has been designed to include all the functions most commonly needed by the home user. For example it has insert/overwrite modes, wordwrap, find, word count, block delete/move/copy/print, redefinable centronics control codes (30), 16K text space (blank lines only occupy 1 character space), lines of *ANY* length from one character to the full text length, margins, thirty-one setable TABS, supports both the ZX printer and most centronics printers (with built-in software for the Kempston interface) and RAM! file facilities for the 128K Spectrum.

To LOAD the program use LOAD "" (48K/+) or select the Tape Loader option on the 128K Spectrum. The program loads in two sections and will display the copyright screen when loading is complete. The program will recognise automatically which version of the Spectrum is being used. Press any key to start.

Specword is menu driven and users of the 128K Spectrum will have no trouble in using the menu system as it is exactly the same as the Sinclair start up menus.

MENUS: To select a menu option use the UP/DOWN arrow keys to move the blue select line to your choice then press ENTER. This will either take you to a sub menu or into word processing. This is the main menu and should the program stop due to any input errors then type 'goto menu' to restart from this menu. (The program will error trap most illegal inputs but cannot trap those associated with illegal RAM! file names).

This menu has the following options:

1) Create.

Before word processing can begin the text file being created must have a title. Selecting this option will produce a sub menu asking you to confirm your choice or Exit back to the Document menu. If you wish to continue to create a document just press ENTER. You will now be prompted to enter a title for your document. This must comprise of valid ascii characters only (no keywords).

The title given to your document will stay with the document at all times unless the retitle option is used (see Save). The title will appear on the top line of your text for reference.

Once a title has been given and accepted you will enter the word processing mode (see Word Processing).

2) Load.

This will give a sub menu for Loading next files.

i) Tape

This option will load a text file from tape (the BASIC program can be modified to allow M/drives etc. to be used).

ii) RAM!

Owners of 128K Spectrums have the option of Loading a text file that has been Saved to a RAM! file. This option will

Listing

```
ZX Spectrum HEXDUMP
SPECWORD 48/+/128K LIST 3
44800  30 33 44 44 44 44 33 00 = A6
44808  00 20 A0 A0 A0 A0 3A 00 = DA
44816  00 45 45 45 45 45 75 00 = CE
44824  00 2E A8 AC 68 68 2E 00 = 80
44832  00 33 44 44 47 44 34 00 = 7A
44840  00 38 A4 A4 B8 A0 A2 00 = 7A
44848  00 74 46 65 44 44 74 00 = 1B
44856  00 49 D5 55 55 55 49 00 = 66
44864  00 9C 50 50 50 50 9C 00 = 80
44872  00 3A 43 5A 4A 4A 3A 00 = A5
44880  00 24 6A AA 2A 2A 24 00 = B0
44888  00 CE A8 AC A8 A8 CE 00 = 40
44896  00 7E 42 42 42 42 7E 00 = 04
44904  00 00 00 20 00 00 00 00 = 20
44912  00 00 00 00 00 00 00 00 = 00
44920  00 00 00 00 00 00 00 00 = 00
44928  00 00 00 00 00 00 00 00 = 00
44936  00 00 00 00 00 00 00 00 = 00
44944  00 00 00 00 00 00 00 00 = 00
44952  00 00 00 00 00 00 00 00 = 00
44960  00 00 00 00 00 00 00 00 = 00
44968  00 00 00 00 00 00 00 00 = 00
44976  00 00 00 00 00 00 00 00 = 00
44984  00 00 00 00 00 00 00 00 = 00
44992  00 00 08 04 7E 04 08 00 = 96
45000  00 00 10 20 7E 20 10 00 = DE
45008  00 00 00 00 00 00 00 00 = 00
45016  00 00 00 00 00 00 00 00 = 00
45024  00 00 00 00 00 00 00 00 = 00
45032  00 00 00 00 00 00 00 00 = 00
45040  00 00 00 00 00 00 00 00 = 00
45048  00 00 00 00 00 00 00 00 = 00
45056  00 00 00 00 00 00 00 00 = 00
45064  00 10 10 10 10 00 10 00 = 50
45072  00 14 14 00 00 00 00 00 = 28
45080  00 14 3E 14 3E 14 00 00 = B8
45088  00 10 3C 50 38 14 78 10 = 70
45096  00 72 54 78 1E 2A 4E 00 = D4
45104  00 20 50 24 58 48 36 00 = 6A
45112  00 10 10 20 00 00 00 00 = 40
45120  00 08 10 10 10 10 08 00 = 50
45128  00 20 10 10 10 10 20 00 = 80
45136  00 24 18 7E 18 24 00 00 = F6
45144  00 08 08 3E 08 08 00 00 = 5E
45152  00 00 00 00 00 10 10 20 = 40
45160  00 00 00 7E 00 00 00 00 = 7E
45168  00 00 00 00 00 10 10 00 = 20
45176  00 02 04 08 10 20 40 00 = 7E
45184  00 3C 46 4A 52 62 3C 00 = BC
45192  00 08 38 08 08 08 3E 00 = 96
45200  00 3C 42 02 3C 40 7E 00 = 7A
45208  00 3C 42 1C 02 42 3C 00 = 1A
45216  00 40 48 48 7E 08 08 00 = 5E
45224  00 7E 40 7C 02 42 3C 00 = BA
45232  00 3C 40 7C 42 42 3C 00 = B8
45240  00 7E 02 04 08 10 20 00 = BC
45248  00 3C 42 3C 42 42 3C 00 = 7A
45256  00 3C 42 42 3E 02 3C 00 = 3C
45264  00 00 10 10 00 10 10 00 = 40
45272  00 00 10 10 00 10 10 20 = 60
45280  00 10 20 40 20 10 00 00 = A0
45288  00 00 7E 00 7E 00 00 00 = FC
45296  00 00 04 02 04 00 00 00 = 1A
45304  00 3C 42 0C 10 00 10 00 = AA
45312  00 3C 5E 56 5E 40 3E 00 = CC
45320  00 3C 42 42 7E 42 42 00 = C2
45328  00 7C 42 7C 42 42 7C 00 = 3A
45336  00 3C 42 40 40 42 3C 00 = 7C
45344  00 7C 42 42 42 42 7C 00 = 00
45352  00 7E 40 7C 40 40 7E 00 = 38
45360  00 7E 40 7C 40 40 40 00 = FA
45368  00 3E 40 5E 42 42 3E 00 = 9E
45376  00 42 42 7E 42 42 42 00 = C8
45384  00 3E 08 08 08 08 3E 00 = 9C
45392  00 1E 04 04 04 44 38 00 = A6
45400  00 44 48 70 48 44 42 00 = CA
45408  00 40 40 40 40 40 7E 00 = BE
45416  00 76 49 49 49 49 49 00 = E3
45424  00 42 62 52 4A 46 42 00 = C8
45432  00 3C 42 42 42 42 3C 00 = 80
45440  00 7C 42 42 7C 40 40 00 = FC
45448  00 3C 42 52 4A 46 3E 00 = 9E
45456  00 7C 42 42 7C 44 42 00 = 02
45464  00 3C 40 3C 02 42 3C 00 = 38
45472  00 3E 08 08 08 08 08 00 = 66
45480  00 42 42 42 42 42 3C 00 = 86
45488  00 42 42 42 42 24 18 00 = 44
45496  00 49 49 49 49 49 36 00 = A3
45504  00 42 24 18 18 24 42 00 = FC
45512  00 41 22 14 08 08 08 00 = 8F
45520  00 7E 04 08 10 20 7E 00 = 38
45528  00 1C 10 10 10 10 1C 00 = 78
45536  00 40 20 10 08 04 02 00 = 7E
45544  00 38 08 08 08 08 38 00 = 90
45552  00 08 1C 2A 08 08 08 00 = 66
45560  00 00 00 00 00 00 00 FF = FF
45568  00 1C 20 78 20 20 7E 00 = 72
45576  00 00 1C 02 1E 22 1E 00 = 7C
45584  00 20 20 3C 22 22 3C 00 = FC
45592  00 00 1E 20 20 20 1E 00 = 9C
45600  00 02 02 1E 22 22 1E 00 = 84
45608  00 00 1C 22 3E 20 1E 00 = BA
45616  00 0E 10 3C 10 10 10 00 = 8A
45624  00 00 1E 22 22 1E 02 3C = BE
45632  00 20 20 3C 22 22 22 00 = E2
45640  00 08 00 18 08 08 3E 00 = 6E
45648  00 04 00 0C 04 04 04 38 = 54
45656  00 20 20 24 38 24 22 00 = E2
45664  00 20 20 20 20 22 1C 00 = DE
45672  00 00 36 49 49 49 49 00 = 5A
45680  00 00 2C 32 22 22 22 00 = C4
45688  00 00 1C 22 22 22 1C 00 = 9E
45696  00 00 3C 22 22 3C 20 20 = FC
45704  00 00 3C 44 44 3C 06 04 = 0A
45712  00 00 2C 32 20 20 20 00 = DE
45720  00 00 1C 20 1C 02 3C 00 = 96
45728  00 10 3C 10 10 12 0C 00 = 8A
45736  00 00 22 22 22 26 1A 00 = A6
45744  00 00 22 22 22 14 08 00 = 82
45752  00 00 49 49 49 49 36 00 = 5A
45760  00 00 22 14 08 14 22 00 = 74
45768  00 00 22 22 22 1E 02 3C = C2
45776  00 00 3E 04 08 10 3E 00 = 98
45784  00 10 10 20 10 10 00 00 = 70
45792  00 00 00 00 00 00 00 00 = 40
45800  10 08 08 04 08 08 10 00 = 44
```

45808	00	14	28	00	00	00	00	00	=	3C
45816	00	10	10	00	00	00	00	00	=	20
45824	FF	FE	FC	F8	F0	E0	C0	80	=	01
45832	FF	FE	FC	F8	F0	E0	C0	80	=	01
45840	00	28	28	00	00	00	00	00	=	50
45848	00	08	08	10	00	00	00	00	=	20
45856	00	10	10	00	00	00	00	00	=	20
45864	FF	FE	3C	42	3C	02	7C	00	=	35
45872	00	00	00	0F	08	08	08	08	=	2F
45880	00	00	00	00	00	00	00	00	=	00
45888	00	00	00	00	00	00	00	00	=	00
45896	00	00	00	00	00	00	00	00	=	00
45904	00	00	00	00	00	00	00	00	=	00
45912	00	00	00	00	00	00	00	00	=	00
45920	00	00	00	00	00	00	00	00	=	00
45928	00	00	00	00	00	00	00	00	=	00
45936	00	00	00	00	00	00	00	00	=	00
45944	00	00	00	00	00	00	00	00	=	00
45952	00	6C	FE	FE	7C	38	10	00	=	2C
45960	00	10	38	7C	7C	38	10	00	=	88
45968	00	00	00	00	00	00	00	00	=	00
45976	00	08	1C	1C	3E	08	08	00	=	8E
45984	00	00	00	00	00	00	00	00	=	00
45992	00	00	00	F8	08	08	08	08	=	18
46000	08	08	08	0F	08	08	08	08	=	47
46008	08	08	08	F8	08	08	08	08	=	30
46016	00	00	00	FF	08	08	08	08	=	1F
46024	08	08	08	0F	00	00	00	00	=	27
46032	08	08	08	F8	00	00	00	00	=	10
46040	00	00	00	00	00	00	00	00	=	00
46048	08	08	08	08	08	08	08	08	=	40
46056	00	00	00	FF	00	00	00	00	=	FF
46064	08	08	08	FF	00	00	00	00	=	17
46072	08	08	08	FF	08	08	08	08	=	37
46080	00	00	00	00	00	00	00	00	=	00
46088	00	00	00	00	00	00	00	00	=	00
46096	00	00	00	00	00	00	00	00	=	00
46104	00	00	00	00	00	00	00	00	=	00
46112	00	00	00	00	00	00	00	00	=	00
46120	00	00	00	00	00	00	00	00	=	00
46128	00	00	00	00	00	00	00	00	=	00
46136	00	00	00	00	00	00	00	00	=	00
46144	00	00	00	00	00	00	00	00	=	00
46152	00	00	00	00	00	00	00	00	=	00
46160	00	00	00	00	00	00	00	00	=	00
46168	00	00	00	00	00	00	00	00	=	00
46176	00	00	00	00	00	00	00	00	=	00
46184	00	00	00	00	00	00	00	00	=	00
46192	00	00	00	00	00	00	00	00	=	00
46200	00	00	00	00	00	00	00	00	=	00
46208	00	00	00	00	00	00	00	00	=	00
46216	00	00	00	00	00	00	00	00	=	00
46224	00	00	00	00	00	00	00	00	=	00
46232	00	00	00	00	00	00	00	00	=	00
46240	00	00	00	00	00	00	00	00	=	00
46248	00	00	00	00	00	00	00	00	=	00
46256	00	00	00	00	00	00	00	00	=	00
46264	00	00	00	00	00	00	00	00	=	00
46272	00	00	00	00	00	00	00	00	=	00
46280	00	00	00	00	00	00	00	00	=	00
46288	00	00	00	00	00	00	00	00	=	00
46296	00	00	00	00	00	00	00	00	=	00
46304	00	00	00	00	00	00	00	00	=	00
46312	00	00	00	00	00	00	00	00	=	00
46320	00	00	00	00	00	00	00	00	=	00
46328	00	3C	42	0C	10	00	10	00	=	AA
46336	00	3C	5E	56	5E	40	3E	00	=	CC
46344	FF	C3	BD	BD	81	BD	BD	FF	=	36
46352	FF	83	BD	83	BD	BD	83	FF	=	BE
46360	FF	C3	BD	BF	BF	BD	C3	FF	=	7C
46368	FF	83	BD	BD	BD	BD	83	FF	=	F8
46376	FF	81	BF	83	BF	BF	81	FF	=	C0
46384	FF	81	BF	83	BF	BF	BF	FF	=	FE
46392	FF	C1	BF	A1	BD	BD	C1	FF	=	5A
46400	FF	BD	BD	81	BD	BD	BD	FF	=	30
46408	FF	C1	F7	F7	F7	F7	C1	FF	=	5C
46416	FF	E1	FB	FB	FB	BB	C7	FF	=	52
46424	FF	BB	B7	8F	B7	BB	BD	FF	=	2E
46432	FF	BF	BF	BF	BF	BF	81	FF	=	3A
46440	FF	89	B6	B6	B6	B6	B6	FF	=	15
46448	FF	BD	9D	AD	B5	B9	BD	FF	=	30
46456	FF	C3	BD	BD	BD	BD	C3	FF	=	78
46464	FF	83	BD	BD	83	BF	BF	FF	=	FC
46472	FF	C3	BD	AD	B5	B9	C1	FF	=	5A
46480	FF	83	BD	BD	83	BB	BD	FF	=	F6
46488	FF	C3	BF	C3	FD	BD	C3	FF	=	C0
46496	FF	C1	F7	F7	F7	F7	F7	FF	=	92
46504	FF	BD	BD	BD	BD	BD	C3	FF	=	72
46512	FF	BD	BD	BD	BD	DB	E7	FF	=	B4
46520	FF	B6	B6	B6	B6	B6	C9	FF	=	55
46528	FF	BD	DB	E7	E7	DB	BD	FF	=	FC
46536	FF	BE	DD	EB	F7	F7	F7	FF	=	69
46544	FF	81	FB	F7	EF	DF	81	FF	=	C0
46552	00	00	00	00	00	00	00	00	=	00
46560	00	00	00	00	00	00	00	00	=	00
46568	00	00	00	00	00	00	00	00	=	00
46576	00	00	00	00	00	00	00	00	=	00
46584	00	00	00	00	00	00	00	00	=	00
46592	00	00	00	00	00	00	00	00	=	00
46600	FF	FF	E3	FD	E1	DD	E1	FF	=	7C
46608	FF	DF	DF	C3	DD	DD	C3	FF	=	FC
46616	FF	FF	E1	DF	DF	DF	E1	FF	=	5C
46624	FF	FD	FD	E1	DD	DD	E1	FF	=	74
46632	FF	FF	E3	DD	C1	DF	E1	FF	=	3E
46640	FF	F1	EF	C3	EF	EF	EF	FF	=	6E
46648	FF	FF	E1	DD	DD	E1	FD	C3	=	3A
46656	FF	DF	DF	C3	DD	DD	DD	FF	=	16
46664	FF	F7	FF	E7	F7	F7	C1	FF	=	8A
46672	FF	FB	FF	FB	FB	FB	FB	C7	=	AC
46680	FF	DF	DF	DB	C7	DB	DD	FF	=	16
46688	FF	DF	DF	DF	DF	DD	E3	FF	=	3A
46696	FF	FF	C9	B6	B6	B6	B6	FF	=	9E
46704	FF	FF	D3	CD	DD	DD	DD	FF	=	34
46712	FF	FF	E3	DD	DD	DD	E3	FF	=	5A
46720	FF	FF	C3	DD	DD	C3	DF	DF	=	FC
46728	FF	FF	C3	BB	BB	C3	F9	FB	=	EE
46736	FF	FF	D3	CD	DF	DF	DF	FF	=	3A
46744	FF	FF	E3	DF	E3	FD	C3	FF	=	62
46752	FF	EF	C7	EF	EF	ED	F3	FF	=	72
46760	FF	FF	DD	DD	DD	D9	E5	FF	=	52
46768	FF	FF	DD	DD	DD	EB	F7	FF	=	76
46776	FF	FF	B6	B6	B6	B6	C9	FF	=	9E
46784	FF	FF	DD	EB	F7	EB	DD	FF	=	84
46792	FF	FF	DD	DD	DD	E1	FD	C3	=	36
46800	FF	FF	C1	FB	F7	EF	C1	FF	=	60
46808	00	00	00	00	00	00	00	00	=	00
46816	00	00	00	00	00	00	00	00	=	00
46824	00	00	00	00	00	00	00	00	=	00
46832	FF	81	81	81	81	81	81	FF	=	04
46840	FF	81	81	81	81	81	81	FF	=	04
46848	00	00	00	00	00	00	00	00	=	00
46856	00	44	44	44	44	00	44	00	=	54
46864	00	55	55	00	00	00	00	00	=	AA
46872	00	00	00	00	00	00	00	00	=	00
46880	00	00	00	00	00	77	77	00	=	EE

46888	00	77	77	77	77	77	77	00	=	CA
46896	00	00	00	00	00	00	00	00	=	00
46904	00	00	00	00	00	00	00	00	=	00
46912	00	00	00	00	00	00	00	00	=	00
46920	00	00	00	00	00	00	00	00	=	00
46928	22	22	22	22	22	22	22	22	=	10
46936	00	00	00	00	00	00	00	00	=	00
46944	00	00	00	00	00	22	44	00	=	66
46952	00	00	00	00	EE	00	00	00	=	EE
46960	00	00	00	00	00	00	22	00	=	22
46968	00	00	00	00	00	00	00	00	=	00
46976	00	77	55	55	55	55	77	00	=	42
46984	00	66	22	22	22	22	77	00	=	65
46992	00	77	11	11	77	44	77	00	=	CB
47000	00	77	11	33	11	11	77	00	=	54
47008	00	44	55	55	77	11	11	00	=	87
47016	00	77	44	77	11	11	77	00	=	CB
47024	00	77	44	77	55	55	77	00	=	53
47032	00	77	11	11	22	22	22	00	=	FF
47040	00	77	55	77	55	55	77	00	=	64
47048	00	77	55	55	77	11	77	00	=	20
47056	00	00	00	22	00	22	00	00	=	44
47064	00	00	00	22	00	22	44	00	=	88
47072	00	00	60	00	00	00	00	00	=	00
47080	00	00	00	00	00	00	00	00	=	00
47088	00	00	00	00	00	00	00	00	=	00
47096	00	00	00	00	00	00	00	00	=	00
47104	00	00	00	00	00	00	00	00	=	00
47112	00	77	55	77	55	55	55	00	=	42
47120	00	77	55	66	55	55	77	00	=	53
47128	00	77	44	44	44	44	77	00	=	FE
47136	00	66	55	55	55	55	66	00	=	20
47144	00	77	44	77	44	44	77	00	=	31
47152	00	77	44	77	44	44	44	00	=	FE
47160	00	77	44	44	55	55	77	00	=	20
47168	00	55	55	77	55	55	55	00	=	20
47176	00	77	22	22	22	22	77	00	=	76
47184	00	33	11	11	11	11	77	00	=	EE
47192	00	55	55	66	55	55	55	00	=	0F
47200	00	44	44	44	44	44	77	00	=	CB
47208	00	55	77	77	55	55	55	00	=	42
47216	00	77	55	55	55	55	55	00	=	20
47224	00	77	55	55	55	55	77	00	=	42
47232	00	77	55	55	77	44	44	00	=	20
47240	00	77	55	55	55	77	33	00	=	20
47248	00	77	55	55	66	55	55	00	=	31
47256	00	77	44	77	11	55	77	00	=	0F
47264	00	77	22	22	22	22	22	00	=	21
47272	00	55	55	55	55	55	77	00	=	20
47280	00	55	55	55	55	55	22	00	=	CB
47288	00	55	55	55	77	77	55	00	=	42
47296	00	55	55	22	22	55	55	00	=	98
47304	00	55	55	55	22	22	22	00	=	65
47312	00	77	11	22	22	44	77	00	=	87

display a CAT! of RAM files and ask for your choice. 48K/+ owners will be told that this option is not available on their computer should they attempt to select it.

iii) Exit.

Return to the Document menu.

3) Save.

This option gives a sub menu for Saving text files.

i) Tape.

Save a text file to Tape using the title as the name for saving.

ii) RAM!

Save the text file to RAM! Again this is only available on the 128K Spectrum.

NOTE: All text files will be saved with their own parameters intact. That is if for example the right margin has been set and insert mode is off then this will be saved with the text so that there is no need to reset them when reloading the file.

iii) Retitle.

Should you wish to retitle your program before saving it then this option will allow just that. For example you may have loaded a document called 'textone' and after modifying it want to save it under a different name.

iv) Program.

If you have customized Specword in any way using the various options to do so then the whole program can be saved and verified using this option.

v) Exit.

Return to the Document menu.

4) Print.

This will give a sub menu as follows:

i) All text.

This will print out to your printer all of your text file using the parameters as set up using the 'format' see (iii).

ii) Block.

If you have defined a block of text within the word processor mode then this can be printed using this option.

iii) Format.

This will produce a sub menu which will allow you to set up the format of text to be printed.

a) Lines per page.
b) Space between lines.
c) Margin.
d) Start at line no.
e) Page no. at TOP.
f) Page no. at BOT.
g) Wait between pages.
h) Auto line feed.
i) Exit.

Against each of these parameters will be printed their current default values. To alter a parameter use the UP/DOWN arrow keys to place the blue cursor line on your choice then enter the new value. Parameters a) to d) require a number value and will allow inputs of 0 to 999 (after entering your required value press ENTER.

Should you make a mistake then press DELETE to cancel the entry. Parameters e) to h) require Yes/No entry, use the Y and N keys to select your choice.

When you are happy with all the values then select Exit to return to the Print menu.

v) Codes.

This option will allow you to alter the printer codes that can be placed in your text file to control the output of text to a centronics printer. For example <Graphics T> in your text will turn on the compressed print mode and has the codes: 15/255/255/255 allocated to it (each control character must have four codes allocated and any unused codes must be sent to 255).

If your printer requires a different set of control code for compressed characters then use this option to change the set codes. You can even allocate other control functions to the Graphics characters. Once amended the codes can be Saved permanently using 'Save program'. This will produce sub menus to select the Graphics letter required, upper or lower case and finally a list of the codes for the letter selected for you to update.

Each sub menu has an Exit option which will return you to the previous menu.

vi) Initialise.

Should you be using the Kempston Centronics Interface that requires software to drive it then this option will initialise the computer to use the Specword software for printing.

If your interface requires special software then this should be loaded prior to loading Specword.

Note that Specword uses most of the memory of the Spectrum

but that memory between 43000 and 44000 is free for your printer interface software. The ZX printer and similar printers can only be used with the 48K/+ computers and require no additional software to run them, DO NOT use the initialise option when using the ZX printer.

vii) Exit.

Return to the Document menu.

5) Edit.

If a document has been created and a return has been made to the Document menu this this option can be selected to return you to the word processor.

6) CAT!

For 128K users only, this will print a list of text files held in RAM memory.

7) Customize.

The word processing mode of Specword will initially use three different colours to define the various areas of the file as it appears on the screen. The main text will be white ink on a dark blue background, the text entry window will be white ink on a back background and the border will be light blue. Should this combination of colours not be to your liking then you may alter them using this option. New values can be saved permanently using 'Save program'.

Note that when text is saved the combination of colours is saved with it so that when loading back into Specword the colours may be different to those you may be using.

This options also allows you to redefine the £ sign for your printer. The code is initially set to 129 (for the SHINWA printer).

8) ERASE!

Again for 128K owners only, this option will list the files held in RAM and ask which one you want to erase. To skip this option if selected in error just press ENTER to return to the Document menu.

If you enter a filename that is not valid then the program will break with an error report. Should this happen enter 'goto menu' to restart.

Word Processing

Once you have mastered the menu system then you can begin to enter text using the word processor.

Using 'Create' to title your text you will be taken into the word processing mode. The screen will clear and you will see the following information displayed on line one of the screen.

COL:00001 LIN:00001 GM_EM_ CAPS_WW_INS_LMAR:00000 RMAR:00000

COL: will keep a record of the current cursor column.

LIN: will keep a record of the current cursor line.

GM_: will indicate when Graphics mode is selected.

EM_: will indicate when E.MODE is selected.

CAPS_: will indicate if caps lock is on or off.

WW_: will indicate if word wrap is on or off.

INS_: will indicate if insert mode is on or off.

LMAR: will indicate the column value of the right margin if set.

RMAR: will indicate the column value of the left margin if set.

The main text screen will be blank except for a series of dots at the beginning of each line. Towards the bottom of the screen you will see a black line, this is the text window in which all typing is carried out. Above this window there is the text title line to remind you of the name given to your document. Below the window is the end of text marker line. A cursor will be flashing black/white at the beginning of the text window.

Initially your text file is just one character long, a new line character code 13 represented by a dot will be seen under the flashing cursor. All margins and tabs will be reset and insert mode will be on (indicated by the white bar against INS in the status line) and word wrap will be off.

In word processing there are three distinct modes which give access to the various functions.

1) Text Mode

This is the standard mode on entry to word processing in which normal text entry is carried out. Specword has a keyboard buffer to ensure that all key presses are read so that even the fastest typist will not be able to out-type the program.

Use the keyboard as you would a normal typewriter. Typing will always appear in the text window so that you do not have to search the screen to find out where you are. When the cursor reaches column 29 the screen will scroll left to keep text entry in view at all times.

Text entry will continue on line one until ENTER is pressed to indicate a new line is required (see margins). You will notice a dot placed at the end of each line, this indicates a new line character and is part of the text, that is it can be erased to join two lines together or inserted into a line to split it into two separate lines.

In all modes the INV. VIDEO key has a special function in that if pressed it will display a HELP page for that mode. These HELP pages give brief information on all the main functions available in that mode.

Text mode functions

i) Arrows: The cursor can be moved around the text using the four arrow keys. Left and right arrows will move the cursor one character at a time through the text in the relevant directions. The up and down arrows will cause the screen to scroll down and up to place the required text line into the cursor window. When moving up and down through the text the cursor will try to stay in the same column but if the line to be moved into the text window is shorter than that column then the cursor will move to the end of that line (this is similar to program editing on the 128K Spectrum). *Note* that the keyboard is buffered and if the arrow keys are held down too long you will overshoot the required position.

ii) Delete: This key will delete the character behind the cursor and backspace to that position closing up the text.

iii) True Video: Move to the next TAB position (see E.MODE for the method of setting tabs).

iv) Graphics: This is the second mode available and is mainly concerned with the printer control characters (see GRAPHICS MODE).

v) Ext. Mode: The third mode available mainly concerned with the text format and editing (see EXT. MODE).

vi) Edit: Return from word processing to the Document menu.

In text mode all the normal characters are available using CAPS SHIFT and SYMBOL SHIFT. CAPS LOCK will select capitals in the normal way and indicate that it is on by placing a white block against CAPS in the status line.

2) Graphics Mode

This will be of interest if you have a dot matrix printer that has standard controls to select different text formats. Selecting GRAPHICS will place a white block against GM in the status line.

INV VIDEO will produce a HELP page giving the control code letters and their original functions. You will see that for most functions UPPER case is used to switch on the function and lower case to switch it off. There are 30 characters available all of which can be redefined to suit your printer.

Printer control characters do not affect the line length of the text sent to the printer.

There are also eleven form characters that can be printed in this mode. These characters can be used to place borders around your text. The following keys are used and as you will see form a keypad to make them easy to find:

Q	W	E	R	┌	┬	┐	─
A	S	D	F	├	┼	┤	│
Z	X	C		└	┴	┘	

The codes used for these characters are initially for the SHINWA CP80 printer but can be

changed to suit your printer with a few simple POKE commands.

The initial codes are:
┌ 134: ┬ 152: ┐ 149: ‾ 157
├ 150: ┼ 159: ┤ 151: | 156
└ 153: ┴ 158: ┘ 154

Each code has two addresses as follows, the first address is for UPPER case, the second for lower case so in fact you can re-program 22 new characters.

36495/6,150
36530/1,154
36532/3,151
36534/5,149
36536/7,156
36558/9,134
36560/1,157
36562/3,159
36570/1,152
36572/3,158
36576/7,153

CAPS LOCK also works in this mode.

This mode also has a 'text' view which will give a display of the current screen at 128 characters per line. This will be in the form of a graphics display with a letter represented by a line. This can be used to check the format of your text. To obtain this view mode press key 1. The display will remain on until another key is pressed when a return to TEXT mode will be made.

3) Ext. Mode

This mode is entered by pressing the EXTEND MODE key and will be indicated by a white block against EM in the status line.

INV VIDEO will display two HELP pages to give you a summary of the various functions available.

The EXTEND MODE key should not be held down once E. MODE has been selected.

The characters ^/ : \ }{ [ ] are available in this mode and the copyright symbol which is non standard has been replaced by the ' symbol (the symbol ' is available in text mode using SYMBOL SHIFT/7).

i) Arrow keys have a special function in this mode as follows:
UP will take you to the start of the text file.
DOWN will take you to the end of the text file.
LEFT will take you to the left margin or if not set the beginning of the line.
RIGHT will take you to the right margin or if not set the end of the line.

ii) Q (<=) will set the left margin to the current cursor value, and display that value against LMAR in the status line.

iii) E (>=) will set the right margin to the current cursor value, and display that value against RMAR in the status line.

iv) R (<) reset the left margin to zero.

v) T (>) reset the right margin to zero.

vi) I Insert mode ON. This will be indicated in the status line by placing a white block against INS. In this mode characters will be inserted into the text displacing the text to the right to make room for that character.

ENTER in this mode will produce a new line in the text displacing the text down one line to make room. The cursor will flash black/white.

iii) O Overprint mode ON. This will be indicated by a short white block against INS in the status line. The cursor will flash red/white. Text entered in this mode will overprint the current text. If the right margin has been set then this mode will not allow printing to continue on the next line until ENTER is pressed. This has been done to stop over-printing of text on the next line without confirming that it is OK to do so. A reminder that the right margin has been set will be printed on the screen should you attempt to exceed the right margin.

ix) K (+) Word wrap ON. This will be indicated by a long white block against WW in the status line. There will also be a change of cursor colour, the white flash will change to blue. That is:
INSERT ON/ WW ON = BLACK/BLUE
INSERT ON/WW OFF = BLACK/WHITE
OPRINT ON/ WW ON = RED/BLUE
OPRINT ON/ WW OFF = RED/WHITE

Word wrap will only function if the right margin is set and insert mode is on. Should a word continue from the end of a line to the beginning of the next line then that word will automatically be transferred to the next line.

ix) J (—) Word wrap OFF. This is indicated by a short white block against WW in the status line. Words will remain split between the end of one line and the start of the next providing that the right margin has been set.

x) C Count (word) Selecting this function will produce a display of the number of words in your text. This will not include any Graphics printer control code letters or block markers. The count is instantaneous and will remain on screen until a key is pressed.

xi) W Word Find. This will search through your text FROM THE CURSOR POSITION to find a word or phrase of your choice. You will be prompted to enter your search text and Specword will search the file until it finds the required word/phrase and place the cursor at the start of that word. Should the required word/phrase not be in the text then the message 'WORD NOT IN TEXT' will be displayed and the cursor will remain in its original position.

xii) 1 Set TAB. This will set up a tab stop at the current cursor position. The border will flash dark blue to indicate that the command has been accepted. Up to 32 tab stops can be set BUT they must be set with column values of less than 256.

xiii) 2 Reset TAB. This will remove a set tab. The border will flash red to indicate that the tab has been removed.

xiv) 3 Reset all TABS. This command will remove ALL the set tabs and the border will flash green to indicate that the command has been accepted.

xv) 8 Set block start. This command will place a block start marker at the current cursor position. This will be indicated by a forward facing arrow.

xvi) 9 Set block end. This command will place a block end marker at the current cursor position and will be indicated by a backward facing arrow.

Any previously set block markers will be removed before a new block marker is set.

xvii) 0 Reset blocks. This will remove any set block markers from the text.

Block markers can be treated as a character in the text file, that is they can be erased or re-positioned by inserting text into the block. This will have no effect on any of the block functions. Specword will allow for any change in position of these markers.

xviii) 5 COPY BLOCK. This command will copy a set block to the current cursor position opening up the text to make room. A black can be any length from just one character to the whole text file. If there is not enough room left in the text file to make a block copy then a message to that effect will be given and the copy command will be ignored.

The block copy will be free from block markers, these will remain set on the original block.

xix) 6 MOVE BLOCK. This will move a set block from its current position closing up the text, to the current cursor position opening up the text. This command will work even on a full text file. The block will be moved complete with its block markers.

xx) 7 DELETE BLOCK. This final command will erase a set block from the text file closing up the text.

Block commands are almost instantaneous but will not work if:
a) The cursor is placed inside the black to be MOVED or COPIED.
b) The start marker is positioned AFTER the end marker.
c) There is insufficient text room to accept block COPY.

As an example of the text that can be created in a file, these instructions were produced on Specword and took 514 lines of which 406 were produced in one text file (2819 words). The remaining 108 lines were produced in another text file (854 words).

IMPOSSABALL



It's Impossaball! A new dimension added to the bouncing ball puzzle game by Hewson.

Impossaball
Hewson
£8.95

Like all the best games Hewson's Impossaball is simple to play but hard to master. It's one of those games that sort of sneaks up on you after a couple of attempts — you may not get very far at first, but as you come up against each new obstacle you can almost see how to get through, so you go back to the start for one more try and before you know it you're hooked.

The basic idea behind the game is very simple but it's been excellently implemented by programmer John Phillips. You control the movements of a bouncing ball over a scrolling three dimensional obstacle in an attempt to complete the course within a time limit, and at the same time as avoiding the deadly obstacles you also have to 'squash' a series of cylinders



that are scattered over the course and which are, of course, surrounded by some of the more deadly traps and barriers.

There are eight courses in the game, each harder than the last, though even the first of these isn't exactly a push-over. At the start of each game the ball is just behind the starting line for each course and, since the timer doesn't start until you've crossed the line this means that you're given an opportunity to bounce the ball around a bit and get the hang of controlling its movements before you start the game properly — a nice idea, and one which allows you to get started quickly (unlike a lot of games recently, Impossaball doesn't force you to read a 50 page manual before you can get started).

As you cross the starting line you move from right to left across the screen and the chequered landscape of the course scrolls smoothly along to reveal each new set of obstacles.



Poles apart

On the first course this consists of spike-topped poles which jut up from the ground or hang from the ceiling, and plasma bolts (which look like animated Ringos). The cylinders that have to be 'squashed' are column-like structures projecting from the floor and ceiling, and 'squashing' them simply involves bouncing onto them from above or below, forcing them to retract into the surface. On later courses though, there are all sorts of deadly traps (most of which are moving and require expert judgement and planning to get through).

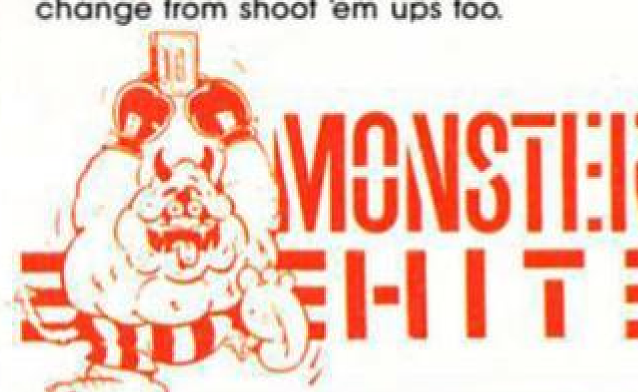
Surprisingly for a game where you're controlling a moving object and racing against the clock, Impossaball doesn't require dazzling arcade-trained reflexes. Despite the time limit there are places where you're better off taking things slowly and thinking your way through an obstacle rather than just barging through as fast as you can. But when the clock's ticking away and you need to hurtle down the home straight to the finishing line, control of the ball is very simple, and the way in which the pace of the game can vary makes it that much more addictive.

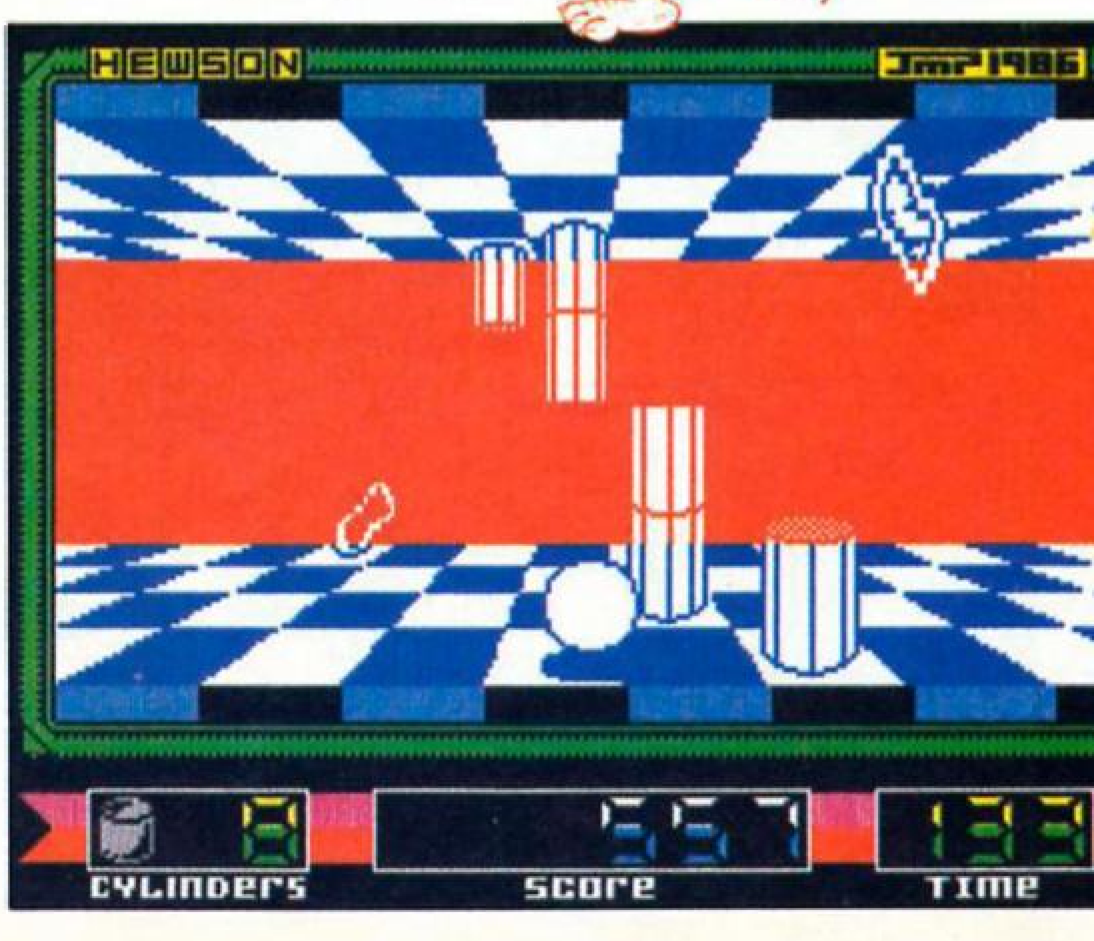
One of the really nice features of the game is that it presents you with each course and then just lets you get on with it. There isn't a single, fixed method of solving each obstacle and there's enough flexibility in the game to allow each player to develop his or her own style of playing. The only suggestion I can make that might improve the playability of the game is that after you've completed a course it might have been a good idea to give you a password that allows you to enter the next level whenever you want. As it is, you've got to go through all the course in a fixed order and even when you've got the first level sorted out you've still got to go through it each time in order to get onto the later ones.

Flicker free

The graphics and *animation are* about as good as I've seen on the Spectrum. All the objects are large and finely drawn, yet the animation is perfectly smooth. There's not the slightest flicker despite the fact that the whole screen display has to swing backwards and forwards very quickly in order to create the impression of bouncing movement. The chequered pattern on the landscape helps with the 3D perspective and also helps when it comes to judging distances and heights of jumps past obstacles, and, as the ball bounces along its shadow follows along, shrinking or growing as the height of the jumps varies.

There have been similar games released recently (Revolution and Action reflex), but Impossaball beats them all in terms of graphics and playability, and it makes a nice change from shoot 'em ups too.





BMX SIMULATOR

A budget BMX racing game that's fun from start to finish

BMX Simulator
Codemasters
£1.99

While BMX racing may not have the same status or glamour as motorcycle or grand prix racing, when it comes to computer simulations BMX comes into its own. Formula one racing simulations are often a bit of a let down, your speedometer may be saying 240 mph but the race action doesn't reflect it, but with BMX racing, a simulation can get closer to the real thing and the intricacy and skill involved required to negotiate the course make it ideal for those who want to use their joystick for more than mindless zapping.

BMX Simulator is an excellent game that is surely destined to become a budget hit. There are seven different courses, seen from an overhead view and these offer a multitude of daunting features becoming harder with each successive track. To qualify for the next track you must complete the course within a given time limit — this is fairly generous for the first course but becomes more stringent with track two.

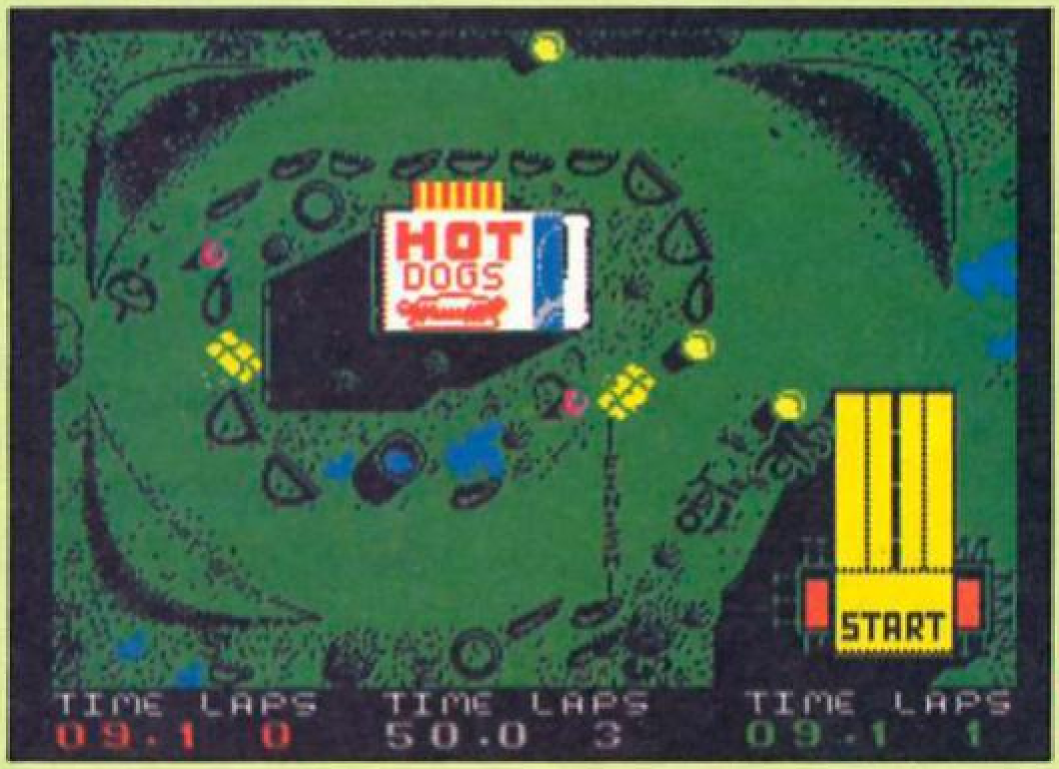
There are options for one or two player games. When playing the computer rider it's gratifying to find that he's very fallible and wipes out fairly often, 'though he does have the advantage in that whenever you collide with him you are always the one to go flying. Crashing is illustrated by a nice animated sequence where your rider and bike part company in spectacular fashion. Restarting the race can be a problem with a delicate touch needed to steer yourself clear of the object you've run into, otherwise you crash again . . . and again . . .

The two player game brings out the full potential of the simulation and most importantly there are options to play with two joysticks (Kempston/Sinclair) this was a wise decision as anyone playing the keyboard option would be at a real disadvantage. Two options have been built in to pep up the after-race debate on who cut up who on the final bend; there is an action replay function which will playback the race, or, if you are really into detailed analysis, you can also watch the whole race again in slow motion. I'm not sure how often these would be used after the novelty has worn off but they are clever additions nonetheless.

Above all, BMX Simulator is very realistic and the bike reacts to the terrain exactly as it would on a real track — the ramps, turns, bumps and water splashes *are not just there for decoration*, they can be used for your advantage.

BMX Simulation will appeal to all those who like race games even if the idea of BMX racing seems a bit tame. Everything about the game's presentation is highly professional from the graphics to course design — another game to add to Codemasters' increasingly impressive track record.







VIEWPOINT

David Nowotnik tests a new 3D graphics package for the QL.

Viewpoint
Rubicon Computer Systems
£19.95

If you have ever been fascinated by the ability of computers to display three dimensional images, then an excellent new software product for the QL will permit you to try out the concepts for yourself. 'Viewpoint' is a machine code program which allows you to construct 3D wire objects, then display the object in two dimensions (the monitor screen). Objects can be viewed from different viewpoints, and you can zoom in and zoom out, change the perspective, achieve some hidden line removal, and save and load object data to and from microdrive or disc.

A printer dump of the screen display is also possible, and **Fig. 1** shows the screen dump on an Epson printer of just one of the example objects supplied with Viewpoint.

The product comprises a single microdrive cartridge and a 20 page 5" x 3" manual. It is demonstration of the simplicity of use of Viewpoint that this small manual more than adequately covers the operation of the program. The program is menu-driven, with options appearing in the menu window at the base of the screen. Five main menus are available, obtained by pressing one of the function keys.

While the program itself is easy to use, 3D graphics, in concept are rather tricky. Here there are new buzz words to understand; world coordinates, spherical polar coordinates, cartesian axes, vectors, angles O and O. Built in to Viewpoint is an option to have some of these terms displayed graphically. This can be done at any time, even while you are constructing an object. Pressing 'F3' and 'c' displays a simple cube in three dimensions, while 'F3' and 'a' displays the axes of a spherical polar coordinate system. Within these training options, you can play around with vector R, and angles O and O. Once you have gained a feel for these, you can return to constructing your own object.

Objects can be built up by drawing lines between two points. A number of 2D (circle, box, polygon) or 3D (sphere) shapes are drawn automatically at the cursor position, with sizes specified by the user. Several drawing aids are included, such as the addition of axes and/or a calibrated grid. Two vertices (coordinates) previously defined can be joined by a simple join command. The mirror image of an object can be produced, with the user defining the plane of the 'mirror'. And, text can be added; this text remains fixed on the screen display, and does not move if the viewing position of the object is changed.

Mirror Images

Pressing 'F2' brings a search facility into play, in which the computer seeks the nearest defined vertex from the current cursor position. The search area is only a 9 x 9 grid of the world coordinates, which, in effect, means that the cursor has to be virtually on the vertex point for it to be found. A bug in the review copy meant that the program crashed if a search was requested when no vertices had yet been defined; this bug the suppliers promise to remove in all future copies. Each line or shape can appear in any user defined colour on Viewpoint's black background. Switching between QL's 4 and 8 colour modes can be accomplished within the program, diagrams being re-drawn to the same scale as soon as the mode has been changed.

In moving the view position relative to the object, the object has to be re-drawn on the screen. With large objects, this can be quite time consuming. In particular, multi-colour displays are re-drawn quite slowly compared to monochromic objects. Therefore, to save time, Viewpoint allows the user to temporarily turn off the multi-colour facility. The resultant red-on-black display can be redrawn much faster.

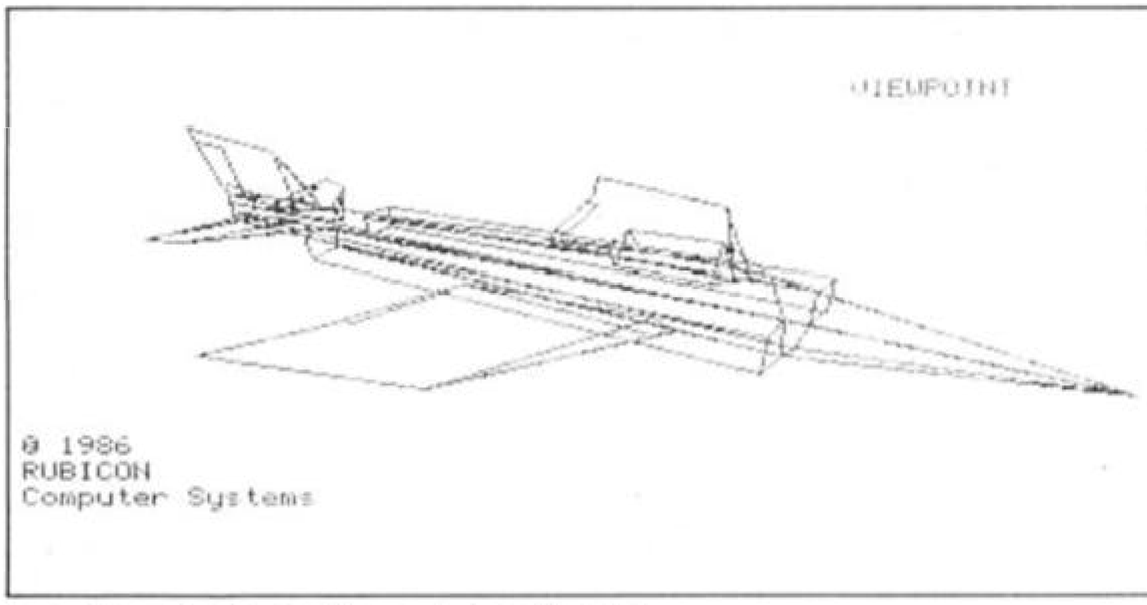
For the more experienced user, Viewpoint has a number of help facilities to aid object construction, such as listing (screen and/or printer) all stored vertices or surfaces of the object under construction.

For a printer screen dump, Viewpoint has the relevant control codes built in for Epson and Seikosha printers.

If you do not have a printer compatible with either of these printers, then the codes relevant to your printer can be entered as a set of 4 blocks. Unfortunately, there is no way of saving these codes, so if you do have to enter special codes, you must do so every time you use Viewpoint. During review, I had a few problems with achieving a printer dump on my Epson RX80. This turned out to be related to my printer's dip switch positions. So my Epson turned out to be (according to Viewpoint) non-Epson compatible! It was a simple matter to cure the fault (changing one switch position), but worth noting if you are using Viewpoint and experience a similar problem.

Overall, Viewpoint is an impressive package, and, at a RRP of £19.95, provides excellent value for money.

Rubicon Computer Systems, 11 Bannerdale Road, Sheffield S7 2DJ.



Screendump of a jet aircraft example from Viewpoint

THE VALUE OF TABLES

Pete Cooke, author of Tau Ceti and Academy, lets us in on some of his machine code secrets.

Listing 1

```
  1 REM lookup table for screen addresses... Check CLEAR xxxxx before running
 10 LET lookup = 65024: REM  #FE00 in hex
 20 FOR n=0 TO 191: REM Number of screen Lines on Spectrum
 30 LET a=INT (n/64): REM 1/3'rds of screen
 40 LET b=n-64*a: REM  lines within a block
 50 LET c=INT (b/8): REM character clocks
 60 LET d=b-8*c: REM line within character block
 70 LET address=16384+2048*a+32*c+256*d
 80 POKE (lookup+2*n),address-256*INT (address/256)
 90 POKE (lookup+2*n+1),INT (address/256)
 95 POKE address,255: REM shows table being generated...delete later!
100 REM poke in the bytes in low high order
110 NEXT n
120 STOP
130 SAVE "ytable"CODE lookup,384
```

One of the most effective techniques in machine code is the lookup table. In Basic lookup tables correspond roughly with single-dimension arrays holding tables of data, for example:—

10 DIM A (20)
20 FOR N=1 TO 20
30 READ A(N)
40 NEXT
50 DATA <whatever you want>

In machine code lookups are much more common as they give a way of evaluating complex functions **without** writing masses of complex code. Two good examples of the use of lookup tables that spring to mind are SIN tables and SCREEN ADDRESSES.

Screen Addresses

One of the biggest hurdles to graphics programming on the Spectrum is the awful layout of the display file. Anyone watching a SCREEN$ loading will have seen that the display lines are stored in a non-sequential fashion and a great deal of time and effort is needed to write a fast and efficient routine to work out the screen address of any line.

A much simpler way is to generate a lookup of screen addresses in memory. I normally use a short BASIC program to do this. **(Listing 1).**

As an additional check that the program is working the line:—

95 POKE ADDRESS,255

fills the first byte of each line of the screen as the program calculates it's address.

When the program has finished type GOTO 130 to save the table.

I normally store a table like this as high in memory as possible but if your assembler is stored at a high address then change the value lookup to somewhere safe below it (it's best to use a multiple of 256 for reasons explained later).

Having got this table into memory screen addressing now becomes much easier. To get a screen byte address we only need a short segment of code like the one in **(Listing 2).**

In fact we can do even better than this by ensuring the table starts at a multiple of 256 bytes. In that case the low byte of the table's address is always 0 **(Listing 3).**

Although this looks as long as the first version there are several improvements. First the DE register pair is not used. Second most of the instructions are single byte instructions which operate much faster and take up less space on a Z80.

Finally here are 2 simple examples using the Ytable lookup, the first is a fancy clear screen (this won't touch the attributes) **(Listing 4)** and the second is a routine to pixel scroll a window upwards **(Listing 5).**

The beauty of lookup tables for screen addresses is that, with the table in memory, graphics routines become much more straightforward and dozens of routines in a long program can use the same table.

Listing 2

```
                ORG   #C000
                ENT   #C000
    YTABLE EQU   #FE00 ;Whatever lookup is
    :
    :
    :Get Byte Address
    :Entry B=Line
    :     C=Column
    :Exit  HL=Address
    :
    GETADD LD    L,B
           LD    H,0
           ADD   HL,HL ;2 bytes in each table value
           LD    DE,YTABLE ;Base of the table
           ADD   HL,DE ;From offset into table
           LD    A,(HL) ;Pick up low byte
           ADD   A,C ;Add on column offset
           LD    E,A ;Into low byte of DE
           INC   HL ;Point HL to high byte
           LD    D,(HL) ;Pick it up
           EX    DE,HL ;Transfer to HL
           RET   ;Done
    :
```

Listing 3

```
10        ORG   #C000
20        ENT   #C006
30 YTABLE EQU   #FE00 ;Whatever lookup is
40 :
50 ;Get Byte Address (Improved version-YTABLE on page boundary)
60 ;Entry B=Line
70 ;      C=Column
80 ;Exit  HL=Address
90 :
100 GETADD LD   L,B ;B=Screen Line
110        SLA  L ;2 bytes per value (+carry)
120        LD   A,YTABLE/#100
130 ;This is the high byte of the start address of the table
140        ADC  A,0 ;Add any carry from the shift
150        LD   H,A ;Form table offset in HL
160        LD   A,(HL) ;Pick up low byte
170        ADD  A,C ;Add on column offset
180        INC  L ;Now only L needs inc
190        LD   H,(HL) ;Pick it up
200        LD   L,A ;Put low into L so HL=address
210        RET  ;Done
220 :
```

Listing 4

```
10        ORG   #C000
20        ENT   #C000
30 YTABLE EQU   #FE00 :Whatever lookup is
40 :
50 :Example 1.
60 :
70 :Using GETADD for a fancy Clear Screen
80 :(Attributes not touched)
90 :
100 FANCY  LD   B,192 :No of screen lines
110        LD   C,0
120 :
130 F_LOOP PUSH BC :Save the counter
131        DEC  B :As DJNZ only loops to B=1
140        CALL GETADD :HL = address of line
170        LD   BC,31 :One less than no of bytes/line
180 :32 bytes on each line
190        LD   E,L
200        LD   D,H :Copy HL into DE
210        INC  DE :DE is destination for LDIR
220        LD   (HL),0 :Clear the leftmost byte
230        LDIR :And copy into the other 32
240        POP  BC :Get line counter back
250 :
260        HALT :A short delay.
270        DJNZ F_LOOP :Loop until done
280        RET
290 :
300 GETADD LD   L,B
310        SLA  L
320        LD   A,YTABLE/#100
330        ADC  A,0
340        LD   H,A
350        LD   A,(HL)
360        ADD  A,C
370        INC  L
380        LD   H,(HL)
390        LD   L,A
400        RET
410 :
```

Listing 5

```
10        ORG   #C000
20        ENT   #C000
30 YTABLE EQU   #FE00 :Whatever lookup is
40 :
50 :These lines form a
60 :short test for the
70 :routine.
80 :
90        LD   BC,#020B
100       LD   DE,#1010
110       CALL PIXSCR
120       RET
130 :
140 :Pixel Scroll a window
150 :upwards....
160 :
170 :Entry B = ystart
180 :      C = xstart
190 :      D = Depth
200 :      E = Width
210 :
220 :All Parameters in
230 :Character Positions
240 :
250 ;Exit A,BC,HL,IX
260 :     all corrupt
270 :
280 :     DE preserved
290 :
300 PIXSCR EQU  $
310        LD   A,B :Test start y
320        CP   24  :for bottom of screen
330        RET  NC :and abort if too big
340        ADD  A,D :also test base of window
350        CP   24
360        RET  NC :and abort if too big
370 :
380        LD   A,C :similarly test x start
390        CP   32 :screen is 32 chars wide
400        RET  NC :abort if too big
410        ADD  A,E :and check right hand
420        CP   32 :edge of window
430        RET  NC
440 :all the above tests could
450 :be cut out in a finished
460 :program where you are sure
470 :that the window size
480 :is always legal.
490 :
500 :
510        LD   L,B :pick up the y start
520        LD   H,0 :into HL
530        ADD  HL,HL :times it by 8 to
540        ADD  HL,HL :convert to a pixel pos
550        ADD  HL,HL :
560 :
570        ADD  HL,HL :and double this to
580 :index into the ytable
590        EX   DE,HL :switch to DE as no ADD IX,HL
600 :
610        LD   IX,YTABLE
620        ADD  IX,DE :Point IX to the
630 :right place in the ytable
640        EX   DE,HL :get DE back
650 :
660        LD   A,D :now convert character
670        ADD  A,A :depth into number
680        ADD  A,A :of pixel lines
690        ADD  A,A
700        DEC  A :we move one less than this
710        LD   B,A :put this counter into B
720
730 :
740 :Now B = No of pixel
750 :       lines to move
760 :    C = X start pos of
770 :       the window
780 :    E = Width of the
790 :       window
800 :   IX  Points to the
810 :       position in the
820 :       Ytable
830 :
840 :the loop below moves each pixel line up 1
850 PIXSC1 PUSH BC :save these registers
860        PUSH DE :as they are corrupted by
870 :this loop
880        LD   B,E :get width into B for inner loop
890        LD   A,C :get byte position
900        ADD  A,(IX) :of start of this
910        LD   E,A :pixel line to
920        LD   D,(IX+1) :DE reg pair
930        LD   A,C :and byte address
940        ADD  A,(IX+2) :of start of line below
950        LD   L,A :into HL register
960        LD   H,(IX+3) :pair
970 :
980 PIXSC2 LDI  :copy (HL) to (DE)
990        DJNZ PIXSC2 :for width of window
1000 :
1010       INC  IX :step pointer to
1020       INC  IX :next line
1030       POP  DE
1040       POP  BC :restore both reg's
1050       DJNZ PIXSC1 :back to move next line
1060 :
1070       LD   A,C :this is byte address of
1080       ADD  A,(IX) :bottom left hand corner
1090       LD   L,A :of the window
1100       LD   H,(IX+1) :into HL
1110       LD   B,E : width to B again
1120 PIXSC3 LD  (HL),0 :fill bottom line
1130       INC  L :with zero's
1140       DJNZ PIXSC3
1150       RET  :finished....
```

ACROSS THE POND

Mark Fendrick previews America's largest Sinclair show.

Firstly I want to answer those of you who have written to ask why this column has not been appearing every month. This was not because Bryan had asked for less columns, but because of the situation concerning the Sinclair market here in the United States. In short, since Sinclair has left our part of the market, and Amstrad has no interest in promoting Sinclair technology in North America, nothing much has happened. The number of Sinclair dealers has remained about the same — or even had its numbers reduced. New sales of Sinclair computers — mostly the QL — continue to be to current Timex/Sinclair owners who have decided to upgrade from their T/S 2068.

There are still a handfull of new products being developed, but not in sufficient numbers to appeal to new home computer buyers, and lack of IBM compatibility turns most potential customers off. The microdrives which were Sir Clive's personal favorite have proven unreliable and many computers which test out fine by the dealer arrive at the mail order customer in an unworkable condition.

But before you all start thinking that I have given up on the Sinclair computers, let me assure you that nothing is further from the truth. More than three years ago I promised that North American Sinclair computerists will always be able to find support here, and rather than dwelling on what is not, we shall seek out all the new happenings and report on them here in ZX Computing Monthly.

Last year one event proved more of a shot in the arm than any other happening. That event was the first annual Midwest Timex Sinclair Computer Fest hed in Cincinnati, Ohio.

Vendors who attended this event all came away from Cincinnati with the feeling that there still is a great deal of interest amongst the Sinclair computer owners in keeping their units viable. A few new products made their appearance at this show, and a number of extremely well attended lectures rounded out the weekend's schedule. All predictions of the long awaited demise of the Sinclair computer in the States were once again put to rest.

Mid West Fest

While other shows were planned across the country, none ever got off the drawing board, but the Second Annual Midwest Timex Sinclair Computer Fest has been announced.

Frank Davis, one of the organizers of last year's smashing success has informed me that this year's gathering will be even bigger and offer more to the dealers and end users than the first fest.

The site for the 1987 affair will be the Holiday Inn North in Indianapolis, Indiana. This location has many improvements over last itme. Since many attendees came by way of air (more than had ever been expected) the long drive from the airport cost many more for the taxi ride than the airfare. The Holiday Inn is not only much closer to the airport, but the hotel supplies a shuttle services from the airport.
square feet of show space dedicated to the show as opposed to the 2400 which was taken up in Cincinnati. There will be tables available for vendors and users groups so that all possible aspects of the Sinclair computing community will once again be represented.

Because the lectures and discussions proved so very popular last year, two rooms have been reserved for this purpose. The plans call for the discussions which proved popular to be repeated at various times throughout the course of the weekend. (Once again I have been asked to lead one or two of these discussions. I hope to have the chance to meet many of you at that time.)

One of the more popular events which could have used more time last year was a swap meet in which participants could swap any form of software/hardware which they wanted. As last year, arrangements have been made to assure that there will be no illegal copying of copyright software. Those coming to Indianapolis should be prepared to bring those items which have been gathering dust, and possibly may be turned into something you could use. This may be the opportunity you have been waiting for to obtain that product you never purchased before it was taken off the market.

The date for this once a year event is the weekend of May 2nd and 3rd.

For more information concerning the show itself, including rates for tables for dealers, absentee and user groups, contact show organizer Frank Davis; 513 East Main Street; Peru Indiana 46970. Please include a SASE to help defray the costs of postage. I hope to see all of you there in Indianapolis.

I have a number of new software packages which have arrived recently which I will report on in the next few columns. Among the programs are music composition for the T/S 2068, a number of new telecommunications products for the entire product line, and a step in the direction of desktop publishing on the QL. I have been informed that there are a few products which will be released in the next few months, and as soon as I receive my review copies I will give you my impressions.

For those of you who want to let me know about new products that you have heard of, or new applications that you have discovered, let me remind you how to get in touch with me. By mail I can be reached at:

Mark L. Fendrick
Post Office Box 2392
Secaucus, NJ 07094-0992
USA

Electronically I can be found on THE SOURCE (BCA632), CompuServe (76417,3264); DELPHI (MARKFENDRICK); by name on MCI MAIL or via telex (via WUI) 6502163495. (Products for review may be sent to the above address as well — hint).

ULTIMATE GAMES

U.S. Gold's latest compilation is an Ultimate's Greatest Hits collection.

Master Games
U.S. Gold/Ultimate
£9.99

This is the second of the two compilations of Ultimate games released by U.S. Gold, and by far the better of the two. Ultimate almost singlehandedly pioneered the arcade adventure hybrid which has now become so widespread, and the four titles on this compilation read like a history of the genre.

Side one of the tape kicks off with Atic Atac, the first real arcade/adventure, and the game which introduced that mythical hero, The Sabreman, to the gamesplaying public.

You have to search the five floors of a castle in search of the Golden Key. Along the way you have to map the layout of each floor and find the objects that you'll need to overcome various creatures (such as the Frankenstein monster). Atic Atac was also one of the first games to attempt some sort of 3D graphics by giving an overhead plan view of the rooms and objects. This game, like all the others on this compilation, gave birth to a great horde of imitations, games which were graphically similar but which often lacked the addictiveness of the originals. Because of this, Atic Atac may well seem like a familiar style of budget game to people who didn't catch it the first time around, but you shouldn't underestimate its addictiveness even if the format of the game now seems a little dated.

Next on side one is Sabre Wulf. Each time Ultimate produced a good game this was often followed by disappointment if their next game wasn't a huge improvement on its predecessor, and this was the case when Sabre Wulf followed Atic Atac. Both games were very similar except that the rooms in Atic Atac were replaced by a maze-like path through a jungle. However, games which look a lot like Sabre Wulf are still being produced, which gives you an indication of how good it was at the time.

Sabre Wulf



Underwurlde

Flipping over the tape you first load up Underwurlde. Now this really was an influential game: Nodes of Yesod, Starquake and Tremor are all among the hordes of games that have a more than passing resemblance to Underwurlde. Faced with the task of finding your way out of the underworld, you have to find the weapons and magical gems which are scattered around the caverns and which you'll need if you're to guide the Sabreman past the monsters that stand in your way. For the first time Ultimate began to introduce series of obstacles that needed to be manoeuvred past (floating gas bubbles and swinging ropes), and gave the Sabreman the ability to jump so that the game involved more than just shooting monsters and collecting objects. Now you had to stop and think about where you were going and how to get there, and this added an extra element of difficulty to the game that helped make it even more addictive.

At this point you'd expect the fourth and final game to be Knight Lore, which was the next game that Ultimate produced as well as being the fourth in the Sabre Man sequence (and my personal all-time favourite). Unfortunately Knight Lore has already been included on another compilation, so this one jumps forward to Alien 8, the game which came directly after Knight Lore.

Alien 8 is the robot which you control in an attempt to save a space ship from plunging into a star. Aboard the craft are the records and last survivors of a dying planet who put themselves into cryogenic suspension while their ship sought out a new home world. But of course completing your task involves getting past the strange creatures which populate the ship, and finding your way through the deadly traps which occupy the cryogenic chambers.



Alien 8

By the time they produced Alien 8, Ultimate's reputation was at its height. With its two most recent games they had perfected a style of 3D graphics that were revolutionary, and which has more or less become adopted as the 'standard' way of presenting 3D graphics: Movie, Batman, Sweevo's World, Nexus, Prodigy and many other games are simply variations on the format that Ultimate pioneered, and Alien 8 has still to be bettered in its combination of problem solving, difficult obstacles, and detailed graphics.

Sadly, Ultimate never really progressed from this point and while they've produced some good games since they've been unable to get back to the peak that they reached with the games on this collection. Because these games were so influential, people who have only gotten into computing in the last couple of years are likely to be more familiar with the large number of games that have copied Ultimate's originals, but these games are still among the best of their various types, so this compilation is worth getting if you're at all interested in arcade/adventures.



Underwurlde



Atic Atac

ULTIMATE GAMES

GREAT



/////

DEBUGGING ARTIST II

Flummoxed by the bugs in Artist II? Carol Brooksbank rides to the rescue.

You may have seen in the review of Softechnics' 'Artist II' in the February ZXC, that I mentioned that there are a number of bugs in the program. Certainly, if you have a copy, you will have met some of them.

This is such a good graphics program without the bugs that I decided to see how many I could remove, and now have my copy in good working order. As most of the changes are quite small, I am sure you would like to clean your copies up too, so here is how it is done.

First, a few bugs that I have left alone because their effects are easily corrected or avoided. After using the 'Up' mode to work on the part of the screen normally hidden by the icon menu, you will sometimes find that a couple of lines from the bottom of the screen are transferred to the top when you return to the 'Down' mode. This is easily corrected by using the 'Window' option to scroll the whole screen. When using the 'Enlarge' mode to work in detail on small sections of the screen, the program may crash if you run the cursor off the right hand edge of the enlarged window. Avoid running the cursor off the edge. When you need to work at the right hand side of the enlarged area, move the window over a bit so that the area comes to the centre of the window. If you are on the right hand edge of your picture, move the cursor slowly so that you can control it, and keep it inside the window.

I mentioned the save/load bug in the type font design option in the review. The 'Save' option saves the wrong block of code, and 'Load' crashes the program. Follow this procedure to save/load successfully.

Select the save/load option, and give the file name when prompted. Use the BREAK key to return to BASIC. (Do not start the tape if saving). Enter as a direct command.

LET B=number

number=62268 for font 3
61500 for font 4
63036 for font 5

Enter **GOTO**

72 to load from microdrive/disc
74 to load from tape
82 to save to microdrive/disc
84 to save to tape

Proceed as usual to save/load.

```
BA62 E5                  PUSH HL
BA63 C5                  PUSH BC
BA64 F5                  PUSH AF
BA65 CD7BBA    CHKBUS    CALL BUSY
BA68 20BF                JR NZ, CHKBUS
BA6A F1                  POP AF
BA6B 01BFFF              LD BC, FFBF
BA6E ED79                OUT (C),A
BA70 0EBB                LD C,BB
BA72 ED79                OUT (C),A
BA74 0EBF                LD C,BF
BA76 ED79                OUT (C),A
BA78 C1                  POP BC
BA79 E1                  POP HL
BA7A C9                  RET
BA7B 01BFFF    BUSY      LD BC,FFBF
BA7E ED78                IN A,(C)
BA80 E6C0                AND C0
BA82 C9                  RET
```

Routine to allow use of the Kempston E interface.

The most serious bugs are in the printing operations, in both the main program and the Pagemaker. The following changes should be made before saving a back-up customised copy.

Kempston E

The last command on LINE 97 of the main program BASIC is **COPY : REM CHR$ NOT PI**

My interface is not initialised by this. I am not sure whether this is a quirk of my particular interface, or whether it is always the case that NOT PI is not recognised. If your interface is not initialised after responding 'Y' to 'KEMPTON PRINT?', then change the last command on LINE 97 to **COPY : REM CHR$ 0**

Enabling the screen dumps

The shaded screen dumps work, but the ordinary large and small ones do not if you are using Epson or compatible printers. For the technically minded, this is because there is a mistake in the Epson bit-image graphic codes. The program makes use of the ESC * control code, which allows the bit image mode to be selected by a number which follows "*". This number can be 0-4 or 6, or 5 on some of the latest models. The program uses 7. A call to Epson confirmed by suspicion that there is no such mode, so the fault can be corrected by POKEing a valid number into the locations concerned. You can experiment to see the effect of the various numbers, but I found 1 the most satisfactory. The only problem is that the large screen dump does not quite fit on the page — about two character squares are cut off the right hand edge — but none of the other modes produces a large dump which comes anywhere near to fitting the page.

These are the POKES you need:

POKE 65180,1 (Small screen dump)
POKE 65186,1 (Large screen dump)

You could try juggling with the values of the two pairs of bit-image data. I could not find a more satisfactory combination than those in the program, but if you have more patience than I, the locations and present values are
65181 (0) (Small dump)
65182 (2)
65187 (0) (Large dump)
65188 (4)

The main program can now be saved. As I have no Opus disc unit, I changed line 96 to make the program auto-run.

96 PAPER 7:INK 0:BORDER 0: CLS:CLEAR 25631:LOAD "art2" CODE

SAVE "artistII" LINE 96:SAVE "art2" CODE 26000,39536.

Pagemaker

The separate Pagemaker has the same bit-image error. If you have a Kempston E interface, ignore the instructions in the handbook for making a customised copy, and proceed as follows.

Load the Pagemaker from the tape. As soon as the instructions for customising appear on screen, stop the tape and use the BREAK key. (i.e. before the blocks of code are loaded). Change the number of LINE 20 to LINE 21. Add

20 COPY:REM CHR$ 0

Enter **GO TO 20,** start the tape, and the blocks of code will be loaded. When the whole program is loaded, use the BREAK key to go to BASIC, and enter **POKE 64034,1.**

(You cannot juggle with the bit-image data in this program as they are transmitted by loading A with a number, followed by RST 10H. To change the 0 data, you would need to replace a one-byte instruction, XOR A, with a two-byte one. The data in the main program is held in a table of values which are easily changed).

You can now save a back-up copy of the program with:

SAVE "pagemaker" LINE 20: SAVE "fonts" CODE 29696,3072: SAVE "scrcode" CODE 47000,1275: SAVE "scrdump" CODE 64000,107: SAVE "write-load" CODE 39700,45.

Your customised Pagemaker will work perfectly, except that you must remember when making The Writer text file, the maximum number of characters per line must be 60, not 64 as stated in the handbook. A full Artist II screen still covers 32 characters, so you will need to take account of the fact that a screen takes slightly more than half the width of the page.

If you wish to use the Pagemaker, you will have a copy of The Writer, and those with a Kempston E interface will also have found that this does not work. This is because, despite the handbook, the code in The Writer is for Kempston S. To get a Kempston E going with it, you need the bit of code in figure 1, and to load it into the program at 47714.

I don't normally think it is a reviewer's job to debug commercial software, but this suite is so useful that it is worth the time and effort to get the programs working properly.

READ ONLY

Spectrum +2 Machine Language for the Absolute Beginner
Melbourne House
£8.95

This book contains the best and clearest explanation of the Z80 instructions that I have ever come across. In that respect it is an excellent book for the beginner, but the author seems to have forgotten that the newcomer to machine code also needs to be shown how to string the instructions together to make the computer actually *do* something. The book is peppered with short routines, but most of them simply show that if you load a couple of registers with numbers and add or subtract them, you get the right answers. The beginner might be forgiven for thinking that you could do most of this in half the time with a calculator.

There is an excellent chapter on hex, binary and decimal arithmetic, and it shows clearly how to calculate the conversions from one to the other, which is just as well. Most of the examples are in decimal, there is a decimal loader and no hex loader, but the tables of opcodes and instructions are in hex and there is no conversion table. It seems a certain recipe for disaster to expect the beginner to calculate conversions between conventions.

There is a splendid chapter on interrupts which makes the use of vector addresses absolutely clear, but again the examples only show how to add 1 to a variable on every interrupt, and the chapter ends by saying that 'there are other things you can achieve with interrupt driven routines, but they need some programming experience.'

The author skates briefly over keyboard reading and printing to the screen, but the chapter on sound makes the whole book worthwhile. At last we are given routines which achieve something and show the reader how to build on the examples to make their own routines. This chapter, and the sections on memory paging, will be useful to anyone, even the more experienced machine code user, who is new to the 128K and +2. Despite the title, I think Joe Pritchard is more familiar with the 128K model, because he talks of amplifying the sound via the ear/mic sockets which, of course, the +2 does not have, and there is no mention of the +2's sound socket.

If you are new to the 128K or +2, then get this book. If you are a machine code novice, buy it for its clarity, but get another at the same time with better examples. This one tells you all about Z80 instructions but nothing much about programming.

Carol Brooksbank



BOOK REVIEW

STREAMS AND CHANNELS

If you thought that you couldn't use the ZX Printer with your new 128 then Toni Baker can prove you wrong.

This article contains not one, but two new channels which you can use on your Spectrum. The first is a channel which enables users of the Spectrum 128 (or the 128+2) to be able to use the ZX Printer — or some other compatible printer such as an Alphacom — even in 128K mode. If you don't have a Spectrum 128 then you don't need this new channel of course, because you can use the ZX Printer anyway.

The second is not really a new channel at all, but a modification to an existing channel. It enables the Spectrum to be able to communicate freely with a QL via the Local Area Network available from the ZX Interface One. Of course communication between these computers via the network is already possible, and David Nowotnik has recently been doing an excellent series in ZX Computing on that very task. It is not my intention to duplicate any of his material, but merely to remove a couple of deficiencies in the channel itself. We'll return to this later, meanwhile back to the ZX Printer.

Figure 1

```
IX + 00     (2 bytes):     Z_PRINT      Address of channel "Z" output.
IX + 02     (2 bytes):     REPORT_J     Address of channel "Z" input.
IX + 04     (1 byte):      Z_NAME       Name of channel (ie "Z").
IX + 05     (2 bytes):     Z_IDEN       Constant 1234h identifies new channel.
IX + 07     (2 bytes):     Z_CLOSE      Address of close buffer routine.
IX + 09     (2 bytes):     Z_LEN        Length of channel info (ie 010E).
IX + 0B     (1 byte):      Z_FLAGS      Various flags.

    Bit 7:  Set if leading space not reqd for keywords, reset otherwise.
    Bit 6:  Not used.
    Bit 5:  Not used.
    Bit 4:  Not used.
    Bit 3:  Set if INVERSE status is ON, reset otherwise.
    Bit 2:  Set if OVER status is ON, reset otherwise.
    Bit 1:  Set if exactly two additional parameters required, reset otherwise.
    Bit 0:  Set if exactly one additional parameter required, reset otherwise.

IX + 0C     (1 byte):      Z_XCOORD     X coordinate of print posn in buffer.
IX + 0D     (1 byte):      Z_WIDTH      Width of buffer (ie 20h).
IX + 0E     (100h bytes):  Z_BUFFER     New printer buffer.
```

```
                            ORG B544
DD7E0C      Z_CLOSE         LD A,(Z_XCOORD)           A:= x coordinate within buffer.
A7                          AND A
C8                          RET Z                     Return immediately if buffer empty.

                            ORG B549
DDE5        Z_NEWLINE       PUSH IX
E1                          POP HL                    HL:= address of channel info.
010E00                      LD BC,000E
09                          ADD HL,BC                 HL: points to new printer buffer.
F3                          DI                        Disable interrupts.
0608                        LD B,08
C5          Z_NL_LOOP       PUSH BC
CD73B5                      CALL B573,Z_COPY_LINE     Output next row to ZX Printer.
C1                          POP BC
10F9                        DJNZ Z_NL_LOOP            Output all eight rows.
```

Channel Z

The program as such begins at address B544. If this seems a rather arbitrary address to you then I should explain that the *reason* it starts at B544 is that all the code follows on from the stuff in last month's article (which ran from B000 to B543). Indeed, some of the subroutines developed last month will be used in the routines given here.

The trick to using the ZX Printer on the 128 is to avoid any possibility of erasing or corrupting the OLD printer buffer (addresses 5B00 to 5BFF) — this is because the memory in this range is used by the 128 to store various paging subroutines, system variables, and an alternative machine stack for use when accessing the vast banks of paged memory. If we can arrange things so that we can avoid corrupting this memory then the good old ZX Printer can still work normally. You see, all of the ZX Printer software in the ROM was written before the advent of the 128 and is not compatible with the requirements of the new machine.

All we really need to do then is to create a NEW printer buffer somewhere in memory, and use the new buffer instead of the old

one. The best possible place for this buffer would be as part of the channel information block for the channel — that would mean that the area occupied by the new printer buffer would be reclaimed whenever you closed the channel.

We shall call this new channel "Z", for ZX-Printer. Take a look at Figure One — it shows the structure of the channel information block for the new channel. As you can see, the first eleven bytes store completely standard information, using the standard which was developed throughout this series. (IX+05/06) contains the constant 1234h, which identifies this as being a user-defined channel. This means that it may be opened or closed using some of the software listed last month. (IX+OB), (IX+OC), and (IX+OD) also store information in the same format as the "W" channel given last month — this is so that we can exploit more of last month's subroutines. As you shall see, a collection of new channels is far more advantageous than a single new channel.

Finally, the new printer buffer itself runs from (IX+OE) upwards, and is 100h bytes in length (same as the old printer buffer). The variable Z_XCOORD (IX+OC) will be zero if the printer buffer is completely empty, or non-zero otherwise. If the new buffer is full it will contain 20h. Z_WIDTH (IX+OD) remains constant at 20h — the width of the buffer. This is so that the TAB and comma-control routines from last month may be exploited successfully.

The program begins at the label Z_CLOSE, which is the routine to perform all of the peripheral tasks necessary to close the channel. In fact this merely consists of testing whether or not the buffer is empty, and printing a newline if it isn't.

Newline

The Z_NEWLINE routine may be compared to the ROM routine COPY_BUFF at address OECD. Its purpose is to transfer the contents of the buffer to the ZX Printer itself, before finally erasing the previous buffer contents. The routine Z-EMPTY does the actual erasing. You will notice a couple of differences between the ROM routine COPY_BUFF, and my routine Z_NEWLINE. Firstly, the new printer buffer is used instead of the old one, and secondly the subroutine Z_COPY_LINE is called instead of the original COPY_LINE.

In fact Z_COPY_LINE is itself very similar to the ROM's COPY_LINE routine (at OEF4). The only difference in evidence is

```
3E04                    LD A,04
D3FB                    OUT (FB),A            Switch printer motor off.
FB                      EI                    Enable interrupts.

                        ORG B55F
DDE5      Z_EMPTY       PUSH IX
E1                      POP HL                HL:= address of channel info.
010E00                  LD BC,000E
09                      ADD HL,BC             HL: points to new printer buffer.
AF                      XOR A                 A:= 00.
77        Z_EMPTY_LOOP  LD (HL),A             Zero next byte of buffer.
23                      INC HL                HL: points to next byte in buffer.
10FC                    DJNZ Z_EMPTY_LOOP     Zero entire of buffer.
DDCB0BFE                SET 7,(Z_FLAGS)       Signal "Leading space not required".
DD770C                  LD (Z_XCOORD),A       Reset x coordinate in buffer.
C9                      RET                   Return.

                        ORG B573
78        Z_COPY_LINE   LD A,B                A:= row number to output.
FE03                    CP 03
9F                      SBC A,A               A:= FF (last 2 rows) or 00 (otherwise)
E602                    AND 02                A:= 02 (last 2 rows) or 00 (otherwise)
D3FB                    OUT (FB),A            Switch printer motor on, but with
                                              slow speed for last two rows.
57                      LD D,A                D:= "last 2 rows" flag.
CD541F    Z_CL_LOOP     CALL 1F54,BREAK_KEY   Test BREAK key.
380A                    JR C,Z_CL_NEXT        Jump forward unless BREAK pressed.
3E04                    LD A,04
D3FB                    OUT (FB),A            Switch printer motor off.
FB                      EI                    Enable interrupts..
CD5FB5                  CALL B55F,Z_EMPTY     Empty the new printer buffer.
CF0C                    RST 08/DEFB 0C        Report D, BREAK - CONT repeats.
DBFE      Z_CL_NEXT     IN A,(FE)             A:= status of printer.
87                      ADD A,A
F8                      RET M                 Return if printer not connected.
30EB                    JR NC,Z_CL_LOOP       Wait until printer stylus is ready.
C3120F                  JP 0F12,COPY_L_2A     Jump to print next row from buffer.
                        ORG B594
DD2A4F5C  SEARCH_CH_ALL LD IX,(CHANS)         IX: points to base of channel
                                              information area.
011400                  LD BC,0014
57                      LD D,A                D:= name of channel to search for.
DD09      SCH_CH_LOOP   ADD IX,BC             IX: points to next chan info area.
DD7E00    SEARCH_CH     LD A,(IX+00)
FE80                    CP 80
37                      SCF
C8                      RET Z                 Return with carry set if channel
                                              not found.
DD7E04                  LD A,(IX+04)          A:= name of channel pointed to.
BA                      CP D
C8                      RET Z                 Return with IX pointing to channel
                                              information area, and carry reset,
                                              if search successfull.
DD4E09                  LD C,(IX+09)
DD460A                  LD B,(IX+0A)          BC:= length of channel info.
18EA                    JR SCH_CH_LOOP        Loop back to continue search.

                        ORG B5B2
DD2A515C  Z_PRINT       LD IX,(CURCHL)        IX: points to channel info area.
CD2FB1                  CALL B12F             Deal with keywords, control params,
                                              TAB, and comma control.
F8                      RET M                 Return if tasks complete.
F5                      PUSH AF
283B                    JR Z,Z_GRAPHICS       Jump with graphics characters.
```

```
3033                    JR NC,Z_ASCII           Jump with ASCII characters.
FE0D                    CP 0D
2005                    JR NZ,Z_PR_CTRLS        Jump unless chr is "enter".
CD49B5                  CALL B549,Z_NEWLINE     Print a newline.
187F                    JR Z_EXIT               And jump to exit.
D614       Z_PR_CTRLS   SUB 14
387B                    JR C,Z_EXIT             Jump with controls 00 to 13.
0608                    LD B,08                 B: has bit three set.
2805                    JR Z,Z_INV_OVER         Jump with "inverse control".
3D                      DEC A
0604                    LD B,04                 B: has bit two set.
200F                    JR NZ,Z_AT              Jump unless chr is "over control".
CB19       Z_INV_OVER   RR C
9F                      SBC A,A                 A:= 00 (INV 0 or OVER 0),
                                                 or FF (INV 1 or OVER 1).
DDAE0B                  XOR (Z_FLAGS)           A:= flags byte, but complemented if
                                                parameter is one.
A0                      AND B                   A: all bits reset, except that bit
                                                3 (INV) or bit 2 (OVER) will be
                                                taken from flags byte, and
                                                complemented if parameter is one.
DDAE0B                  XOR (Z_FLAGS)
DD770B                  LD (Z_FLAGS),A          Bit 3 (INV) or bit 2 (OVER) of flags
                                                byte will be assigned with parameter.
1863                    JR Z_EXIT               Jump to exit.
3D         Z_AT         DEC A
2060                    JR NZ,Z_EXIT            Jump unless control is "at control".
79                      LD A,C                  A:= required x coordinate.
FE20                    CP 20
D29F1E                  JP NC,1E9F,REPORT_B     Error if out of range.
DD770C                  LD (Z_XCOORD),A         Assign x coordinate as required.
1855                    JR Z_EXIT               Jump to exit.
ED5B365C   Z_ASCII      LD DE,(CHARS)           DE:= addr of character set -100h.
1811                    JR Z_CHR_1              Jump forward.
D690       Z_GRAPHICS   SUB 90
3009                    JR NC,Z_UDG             Jump with user defined graphics.
47                      LD B,A
CD380B                  CALL 0B38,PO_GR_1       Construct graphic in MEMBOT area.
21925C                  LD HL,MEMBOT            HL: points to graphic form.
180B                    JR Z_CHR_2              Jump forward.
ED5B7B5C   Z_UDG        LD DE,(UDG)             DE:= address of user defined grphcs.
6F         Z_CHR_1      LD L,A
2600                    LD H,00
29                      ADD HL,HL
29                      ADD HL,HL
29                      ADD HL,HL
19                      ADD HL,DE               HL: points to required graphic form.
DD7E0C     Z_CHR_1      LD A,(Z_XCOORD)         A:= current x coordinate in buffer.
FE20                    CP 20
E5                      PUSH HL
CC49B5                  CALL Z,Z_NEWLINE        Print a newline if buffer full.
D1                      POP DE                  DE: points to graphic form.
DD7E0C                  LD A,(Z_XCOORD)         A:= current x coordinate in buffer.
3C                      INC A                   A:= new x coordinate.
DD770C                  LD (Z_XCOORD),A         Store in system variable.
C60D                    ADD A,0D
4F                      LD C,A
0600                    LD B,00                 BC:= displacement to posn in buffer.
DDE5                    PUSH IX
E1                      POP HL                  HL: points to channel information.
09                      ADD HL,BC               HL: points to current posn in buffer.
0608                    LD B,08
EB         Z_CHR_LOOP   EX DE,HL                DE:= current buffer position;
                                                HL: points to graphic form.
```

the effect of pressing the BREAK key — in my routine Z_EMPTY is called instead of CLEAR_PRB. This removes any possible chance of corrupting the old printer buffer area.

A general purpose subroutine is included next — SEARCH_CH_ALL. It is entered with the A register containing the name of a channel. The routine will search through the channel information area looking for a channel with this name. If it finds one it will return with IX pointing to the channel information block and the carry reset, otherwise the carry will be set. We shall make use of this subroutine later.

Z_PRINT

Z_PRINT is the routine which "prints" a character held in the A register onto the new printer buffer. Note that the subroutine at address B12F comes from last month's article and will sort out all keywords, control parameters, and will deal with both the comma-control and the TAB function. On return from this subroutine the C register will contain any INVERSE or OVER parameters, or the x coordinate of any AT parameters. The Z_PRINT routine itself is really quite simple, bearing in mind that it has to deal with ASCII characters, block graphics, and UDG's, as well as OVER, INVERSE, AT and ENTER. Follow it through to see how all the various cases are dealt with.

Finally (for this channel) we have Z_OPEN, which opens the channel and attaches it to a stream. On entry, the A register must contain the stream number to which the channel is to be attached and the subroutine will do the rest. It makes use of another subroutine from last article — the OPEN_NEW routine which will open user defined channels.

Channel Q

When communicating with the QL via the normal network channel you will notice a couple of problems. The first is that on the Spectrum the code for "enter" is OD (thirteen), whereas on the QL the code for "enter" is OA (ten). This means that, for instance, PRINT A$ on the QL cannot be matched by INPUT A$ on the Spectrum, nor vice versa. It is normally necessary to use **PRINT A$;CHR$(13);** on the QL in order that the Spectrum may use INPUT A$. Conversely, INPUT A$ on the QL needs to be matched by PRINT A$; CHR$ 10; on the Spectrum.

What I intend to do is to modify the "N" channel attached to a particular stream so that the deficiencies are removed. In a modified "N" channel these problems disappear. PRINT A$ on the Spectrum may be matched by INPUT A$ on the QL, and vice versa. Other surprising advantages turn up; if you LIST a Spectrum program over a modified network channel then the QL will be able to LOAD the program with the single command LOAD NET1_2 (assuming that the Spectrum is station number 2) — or to transfer the program directly onto a QL microdrive ready for loading later with the single QL command COPY NET1_2 TO MDV1_PROG.

The second problem is a little more complex, but not much so. The Spectrum command INPUT A$ actually expects to receiver a string *expression*, not the contents of an actual string (although this is not the case with INPUT LINE). Fortunately for us, we seldom notice this because the Spectrum provides the surrounding quotes — this converts the string text into a string expression and everything is hunky dory — with one exception! If the string printed over the network by the QL contains a quotes character then the Spectrum will be unable to evaluate the result as a string expression and will halt with error C, Nonsense in BASIC. The cure to this problem is very simple. If, during Spectrum INPUT (as opposed to INKEY$ or INPUT LINE) a quotes character is received from a modified network channel THEN IT MUST BE CONVERTED TO TWO CONSECUTIVE QUOTES CHARACTERS in order that the expression will still evaluate.

Let's take a look at the subroutines now. We begin with SHADOW_DE which will call a subroutine in the Shadow ROM whose address is DE. It achieves this by fooling the Spectrum into thinking that we are *returning* to the Shadow ROM from a Spectrum subroutine, and then simple paging out the Shadow ROM on completion.

Q_PRINT is the routine which "prints" a character over the modified network. As you can see it is extremely simple, merely testing for an "enter" character and converting such to OA, before using the original Shadow-PRINT subroutine to print the character.

Q_INPUT is more complicated, though not much so. Firstly, the routine tests the error return address to see whether the character-input is due to an INPUT command, or due to an INKEY$ function. If it turns out to be merely INKEY$

```
1A                      LD A,(DE)              A:= byte from buffer.
DDCB0B56                BIT 2,(Z_FLAGS)
2001                    JR NZ,Z_CHR_OVER       Jump if OVER off.
AF                      XOR A
DDCB0B5E                BIT 3,(Z_FLAGS)
2801                    JR Z,Z_CHR_INV         Jump if INVERSE on.
2F                      CPL
AE                      XOR (HL)               A:= byte from graphic form, with
                                               OVER and INVERSE taken into account.
12                      LD (DE),A              Store in buffer.
E5                      PUSH HL                Stack pointer into graphic form.
212000                  LD HL,0020
19                      ADD HL,DE              HL: points to appropriate byte in
                                               buffer for next row of character.
D1                      POP DE                 DE: points into graphic form.
13                      INC DE                 DE: points to next byte to use.
10E5                    DJNZ Z_CHR_LOOP        Print whole character into buffer.
F1        Z_EXIT        POP AF                 A:= character just printed.
A7                      AND A                  Reset the carry flag.
C9                      RET                    Return.
                        ORG B64A
08        Z_OPEN        EX AF,AF'              A':= stream number to attach channe
3E5A                    LD A,"Z"
CD94B5                  CALL B594,SEARCH_CH_ALL  Search for another "Z" channel.
D2C415                  JP NC 15C4,REPORT_J    Error if one exists.
DD2144B5                LD IX,Z_CLOSE          IX:= close buffer address.
21B2B5                  LD HL,Z_PRINT          HL:= output address.
11C415                  LD DE,REPORT_J         DE:= input address.
010E01                  LD BC,010E             BC:= length of channel info.

CD6DB0                  CALL B06D,OPEN_NEW     Open the channel.
DD360B00                LD (Z_FLAGS),00        Reset the flags.
DD360D20                LD (Z_WIDTH),20        Specify buffer width = 20h chrs.
C35FB5                  JP B55F,Z_EMPTY        Empty the buffer, and return.

                        ORG B66E
210007    SHADOW_DE     LD HL,0700,UNPAGE
E5                      PUSH HL                Stack UNPAGE address in Shadow ROM.
D5                      PUSH DE                Stack Shadow subroutine address.
65                      LD H,L                 HL:= 0000.
E5                      PUSH HL                Stack 0000 signalling "Return
                                               to Shadow ROM address".
C30800                  JP 0008                Jump to call Shadow subroutine.

                        ORG B678
DD2A515C  Q_PRINT       LD IX,(CURCHL)         IX: points to channel information.
F5                      PUSH AF                Stack character to print.
FE0D                    CP "enter"
2002                    JR NZ,Q_PRINT_2        Jump unless character is "enter".
3E0A                    LD A,0A                A:= QL's code for "enter".
DD5E05    Q_PRINT_2     LD E,(IX+05)
DD5606                  LD D,(IX+06)           DE:= Shadow output address.
CD6EB6                  CALL B66E,SHADOW_DE    Call network output in Shadow ROM.
F1                      POP AF                 A:= character just printed.
A7                      AND A                  Reset the carry flag.
C9                      RET                    Return.
                        ORG B68F
DD2A515C  Q_INPUT       LD IX,(CURCHL)         IX: points to channel information.
2A3D5C                  LD HL,(ERR_SP)         HL: points to error return address.
5E                      LD E,(HL)
23                      INC HL
56                      LD D,(HL)              DE:= error return address.
217F10                  LD HL,107F,ED_ERROR
A7                      AND A
```

STREAMS AND CHANNELS

```
ED52                    SBC HL,DE
2814                    JR Z,Q_INPUT_2          Jump if in a  INPUT statement.
DD5E07    Q_INKEY$      LD E,(IX+07)
DD5608                  LD D,(IX+08)            DE:= Shadow input address.
CD6EB6                  CALL B66E,SHADOW_DE     Call network input from Shadow ROM.
47                      LD B,A                  B:= character just input.
F5                      PUSH AF                 Stack the flags.
FE0A                    CP 0A
2002                    JR NZ,Q_INKEY$_2        Jump unless this is a QL "enter".
060D                    LD B,0D                 B:= Spectrum "enter" character.
F1                      POP AF                  Restore the flags.
78                      LD A,B                  A:= character to return.
C9                      RET                     Return.
ED7B3D5C  Q_INPUT_2     LD SP,(ERR_SP)          Empty the machine stack down as
                                                far as ED_ERROR.
E1                      POP HL                  Drop the address ED_ERROR.
E1                      POP HL                  HL:= normal error return address ptr.
223D5C                  LD (ERR_SP),HL          Restore pointer to error retn addr.
CDA1B6    Q_INP_LOOP    CALL B6A1,Q_INKEY$      Read next character from network.
3804                    JR C,Q_INP_STORE        Jump if character OK.
28F9                    JR Z,Q_INP_LOOP         Jump if still waiting.
CF07                    RST 08/DEFB 07          Report 8, End of file.
FE0D      Q_INP_STORE   CP 0D
C8                      RET Z                   Return (INPUT is now finished) if
                                                character is "enter".
CD850F                  CALL 0F85,ADD_CHAR_1    Store the character in the INPUT line.
FDCB377E                BIT 7,(FLAG_X)
20EB                    JR NZ,Q_INP_LOOP        Loop back if doing INPUT LINE.
FE22                    CP 22
CC850F                  CALL Z,0F85,ADD_CHAR_1  Double-up quote characters.
18E4                    JR Q_INP_LOOP           Go back for rest of INPUT.

                        ORG B6DA
CD2117    MODIFY_N      CALL 1721,STR_DATA_1    BC:= STRMS data for stream supplied.
78                      LD A,B
B1                      OR C
280C                    JR Z,MOD_ERROR          Error if stream not in use.
2A4F5C                  LD HL,(CHANS)           HL: points to base of channel
                                                information area.
09                      ADD HL,BC               HL: pts to 2nd byte of chan info.
23                      INC HL
23                      INC HL
23                      INC HL                  HL: pts to 5th byte of chan info.
7E                      LD A,(HL)               A:= name of channel.
FE4E                    CP "N"
2802                    JR Z,MOD_QL             Jump only if channel is "N".
CF17      MOD_ERROR     RST 08/DEFB 17          Report O, Invalid stream.
2B        MOD_QL        DEC HL
36B6                    LD (HL),B6
2B                      DEC HL
368F                    LD (HL),8F              Store new input address.
2B                      DEC HL
36B6                    LD (HL),B6
2B                      DEC HL
3678                    LD (HL),78              Store new output address.
C9                      RET

                        ORG B6FC
3E04      Z_DEMO        LD A,04                 A:= stream number to attach.
C34AB6                  JP B64A                 Jump to open channel "Z" and
                                                attach it to stream four.
```

then things are just as simple as for Q_PRINT. The shadow input routine is called to read a character, and if that character is 0A then it is converted to "enter".

If, however, we are dealing with INPUT (as opposed to INKEY$) then the first thing we must do is abandon altogether use of the EDITOR routine in the ROM (which we do by emptying the machine stack as far as the return address from the EDITOR call) and then supplying a new routine to replace EDITOR. We can call Q_INKEY$ as a subroutine to fetch a single character from the network (converting 0A to 0D) and the flags will tell us if the character is valid or not. Then we can call a ROM subroutine ADD_CHAR_1 to actually insert the character into the input line. Quote characters must be inserted twice, not once (except during INPUT LINE). Finally, if an "enter" character is received then we may return, as if from the EDITOR routine, with the finished string expression in the INPUT area of RAM.

MODIFY_N is the routine which will convert a "N" channel to do as its told. On entry the A register must contain the stream number of an existing "N" channel which must have been opened normally. In such a channel (IX+00/01) and (IX+02/03) will normally contain the value 0008 which, if called, will page in the Shadow ROM and then call the *actual* output or input subroutine addressed by (IX+05/06) and (IX+07/08) respectively.

The MODIFY_N subroutine will merely convert these 0008s to the addresses of Q_PRINT and Q_INPUT, leaving the addresses of the Shadow routines unchanged. The Shadow routines will then *only* be called when required by Q_PRINT or by Q_INPUT.

Finally, a five byte demonstration of the "Z" channel. Simply calling Z_DEMO from BASIC will open a new channel "Z", and attach it to stream number four.

Channel R

In the next and final part of this series we shall examine the use of the existing channel "R", which may only ever be used in machine code. We shall also be creating a new channel, also called "R" (R for RAM-disc) which will enable you to create READ and WRITE files in the Spectrum 128's RAM-disc area, just as you can on the microdrives. Till then, keep smiling, and don't take *anything* seriously.

THE ULTIMATE TASWORD?

Tasword +2
Tasman Software
£19.50

At last it's here — Tasword +2 on Discovery disc. many of you a familiar with good old Tasword 2 — a little cumbersome, but very easy to use and thoroughly reliable. You probably also know that Tasword 3 is an upgrade of this, with all sorts of additional features — rapid cursor movement, headers, footers, pagination, word and character count, 128 column width, the list of options seems endless.

More text

So what's different about Tasword +2? The major improvement, as in Tasword 128, is the gigantic text file. Tasword 2 allowed 320 lines, roughly 10 double spaced A4 pages. It didn't matter if they were all space; that 's all you got. Tasword 3 improved on this by using text compression techniques. If there was a lot of space, you got more lines in the file. However, as a penalty, the "file full" marker came up without any warning, and then you often had to delete several lines before you could even get the cursor to the bottom of the text again. I had a number of files full of lists of scientific references, each one incomplete and lacking the title of the article. This is where the "aut" insert mode in Tasword 3 came in so useful; I was able to type the title into the middle of the reference without having to split it all first. However, this meant that I was expanding the file "from the middle outwards". And it suddenly threw up "file full" and jammed solid. I resolved to test my new copy of Tasword +2, so I MERGEd the old Tasword 3 files into it. Well, it's true what they say. It holds an enormous amount of text, around 45-50 double spaced A4 sheets, depending on the rest of the spacing in the text. And there are other diffferences from Tasword 3 or Tasword 128, too.

First of all, a lot of little bugs have been ironed out. That nasty little buglet which gives a lot of vertical lines across the screen when you ask the Opus version of Tasword 3 to load a file which doesn't exist has gone. And that problem Tasword 3 had over sequential printing of text files with ten character titles (it refused) has now disappeared, too. Secondly, there are one or two subtle differences in the control keys. The one which is the most noticeable is the use of the symbol-shift and "i" key for inserting characters or spaces instead of the "y" key as in Tasword 2 or Tasword 3. To make matters worse, in Tasword 3, the combination symbol shift and "i" deletes a letter; one needs to be very careful. the whole thing is a bit of a nuisance until you get used to it. Actually, all the symbol shifted key names have been replaced.

These are very minor grumbles, however, and to compensate, there are one or two further improvements. For instance, the action of symbol-shifted "a" is now instant; no delay for the main menu in this version. Automatic insert also works much more smoothly. And the size of the file! Talk about impressive! It contained my 170 references with no trouble at all! It's beautifully programmed and works like a dream. Tasword 3 files are totally compatible, just merge them in, one after the other; Tasword 2 files are readily converted using the T2T3 program supplied.

If you've got a Discovery disc and you want a wordprocessor, then get Tasword +2 — it's the best there is.

CROSSFIRE

Out In The Cold

Does your Spectrum feel the cold? My +2 does! As soon as I had bought my new toy, I rushed to try it out, switched on, tried to type things in ... Capital N and capital M (or NEXT and PAUSE if K cursor was selected) refused to work, although "n" and "m" were quite all right. Capital A was fine, but STOP (symbol shift and A) also failed. Rushed back with my new toy to Evesham Micros (bless 'em). Just the same problem, but it soon vanished: they had a hot-air fan heater just under the counter; the fault seemed to disappear when the thing warmed up. We tested their new cold stock, which also behaved in the same curious way; for a minute or two, no "M", no "N", then fitfully as it warmed up, then all right ...

Now, "N" and "M" are associated with extra circuits which include several diodes and provide single-key entry comma and full stop; I've modified keyboards in this way in the past. So I guessed that perhaps the diodes had a strong temperature coefficient, and their characteristics changed drastically when the Spectrum was turned on. However, it wasn't too much bother, and I didn't worry too much.

I did notice that for Tasword 2 there were rather similar effects, but not as bad. So, on loading this in, I found that at first capitals M and N were rather jerky, and STOP barely worked until the machine warmed up. However, I got on with more important things ...

Tasword 3 duly arrived on disc — no problems: N and M fine from the word go! Well, that's not unreasonable. Tasword 2 uses interrupt mode 2 to scan the keyboard; Tasword 3 uses interrupt mode 3. "Maybe the beast reacts differently in the two modes" thought I. And once again, got on with other things.

Shortly after, Tasword +2 arrived, also on disc. This time, I was, and am in real trouble. Although it uses interrupt mode 3, just like Tasword 3, but this time in 128K mode, the results are quite different. "M", "N" and "STOP" are fine. But, instead, the "cursor-up" key takes almost an hour to come to fruition! I guess this is a problem which is worse for me than most; I keep my +2 in a cold, cold garret.

Back to Evesham Micros (the long-suffering Cotteridge branch, this time). Further new (and cold!) stock (totally different, much later batch) was tested with their rather smaller power supply, with mine, with my Discovery unit; as many combinations as were possible. No difference, they behaved just the same as mine.

Help! Does anyone know what's going on? Do the old 128's behave in the same curious way?

I guess I'm going to have to keep my +2 warm in future!
John Wase.

Printer Tip

If you have a 128K and a HR-5 printer, entering, either as a direct command or as program lines the commands FORMAT "p";300 and POKE 23391,3002 will enable the printer for LLIST, LPRINT and COPY.

The copy is double height and the —2 eliminates the join between lines. It is also advisable to re-enter or re-run the commands after each job because the first COPY line may (but not always) be a little squashed.
K.S. Schimmel, Nottingham.

Writing Games

I have had a Spectrum 48K+ for five months and looked at several computer magazines.

I picked up a copy of ZX recently and was impressed with the amount of varied programs in the magazine.

What I have in mind is to produce a game of my own and I greatly appreciate the utility programs. May I say again that I find your magazine excellent and I will be placing a regular order.

P.S. Are there are any books you could recommend on writing games?
Ian Barrass, Hull.

Two titles on the ZX shelves that may be of interest are How to Write Spectrum and Spectrum + Games Programs by W. Simister (published by Bernard Babani) which concentrates on writing computer versions of traditional board games and Spectrum Gamesmaster by Kay Ewbank, Mike James and S.M. Gee (Granada Publishing) which shows how to master the rudiments of Frogger and other simple arcade games. — Ed.

RANDOM MEMORY

This month we're moving into the big time! That is as far as the size of illustrations goes. No more little minipics. Now we're talking about full screen illustrations.

But first the bad news. Big pics mean big memory blocks to store them — 6912 bytes (memory spaces) to be precise if you want them in glorious Sinclaircolor. If you're envisaging a "Mugsy"-style comic strip, driven by a relatively short program, this may not matter. Let's assume for the moment it doesn't, and see how we call store and recall the illustrations.

First of all, how many can you store in memory? Answer — 5, with about 6K left for the driver program ("128" owners will obviously do much better than this). Next, how do you store them? In principle what you have to do is to transfer the contents of each byte of the display file (D_FILE) and attributes area (ATTR) where the on-screen pictured is stored into high memory, knowing where you're putting it. You could do this by PEEKing each byte, then POKEing the contents up, but this would be quite slow. The solution is a short machine code routine which uses the instruction LDIR. This stands for load/decrement/increment/ repeat which probably doesn't make you much the wiser. In essence, you set one counter to the number of bytes to be transferred, another to the start of the D_FILE, and a third to the destination address. LDIR does the rest, continually transferring bytes until the first counter is reduced to zero.



Table A

33	0	64	17	80	195	26	254
255	40	46	254	0	40	32	119
35	19	124	254	88	56	239	26
254	0	40	9	119	35	19	124
254	91	56	243	201	19	26	213
22	0	95	25	209	24	239	19
26	213	22	0	95	25	209	24
216	19	26	71	54	255	35	16
251	24	206					

The necessary code is included in **Program 1,** which will make the transfers for you, modify the code to make it work in reverse, then save it along with the picture bytes. Type it in, and let's try it out. If you have some pics ready made with a drawing utility (such as Toni Baker's Light Screen Designer) you could use those. Otherwise the "rainbow" and "logo" SCREEN£s at the beginning of your "Horizon's" tape will be quite satisfactory.

RUN the program, and you'll be asked how many pics you want to save. Let's try two. Answer the prompt for a title with the first pic, then play the tape. The picture will load in more or less instantly and be transferred to high memory, the address of which will be given. Make a note of this, and the POKE number as well. You'll need this information later. Repeat for the second picture, and there you are. You can then save the data for the two pics plus the machine code to drop them back down.

To make use of this routine you will need a subroutine in your program such as:

9999 POKE 65358, h : RANDOMIZE USR 65356 : RETURN

Where you have set variable h to the POKE number you noted for that picture, before you GOSUB 9999. Try it, but don't blink at the wrong moment. You'll miss it!

Memory squeeze

Now all this is very well if you have a "128" or don't need many pics. How can you squeeze the proverbial quart into the Spectrum's pint pot? One way is to have a smaller "quart". In other words, perhaps you would be happy with a two colour drawing (INK and PAPER), which takes up 768 bytes less. You could go further by having only a top third, or two two-thirds screen picture (many adventures use this system, leaving the bottom of the screen clear for text). The REMs given in Program 1 will tell you the changes you will need to make for these variations. If you want to mix various types of illustrations in one program you'll need to POKE 65364 with the appropriate value shown in the line 100 REM before you call the routine.

This technique obviously doesn't help if you really want full screen illustrations, but there is a way around that problem if you're prepared to sacrifice a little speed for a great saving in space. If you type in the following line:

LOAD ""SCREEN£ : FOR f=16384 TO 23295 : PRINT H O;"Address ";f;"holds "; PEEK f : PAUSE 20 : NEXT f

Press ENTER, then LOAD in a picture. You'll see a series of numbers appear at the bottom of the screen as the program PEEKs its way through the D_FILE, and later the ATTR if you wait long enough. You'll notice that the numbers 0 and 255 (and later the permanent attribute; for example 56 if the background is black on white) occur more often than any others. This is because most of a picture is either blank space (0) or inked in (255), whilst much of the attributes area remains unchanged. Knowing this, it is possible to compact the data for a picture by storing, for example, a line of 32 zeros as 0,32. Using this technique even a complex picture, such as the "rainbow" SCREEN£ is compacted to about half its usual length. This is what the compactor routine **(Program 2)** does. Type it in and let's try it out.

RUN the program, and answer

PROGRAM 1

```
   5 REM picsave
  10 INPUT "No. of pics to store
? ";n
  20 POKE 23728,n: LET a=65367-n
*6912-12: REM replace 6912 with
2048 for top third of screen,
4096 for two thirds, 6144 for
D_FILE only
  25 CLEAR a: LET n=PEEK 23728:
LET a=65367-n*6912-11: REM 6912
replaced as above
  28 LET s=a: PRINT "CLEAR ";s-1
;" before LOADing": FOR f=65356
TO 65367: READ i: POKE f,i: NEXT
 f
  30 FOR f=1 TO n: INPUT "Title
of pic ";(f);"?",a$: LOAD *"m";1
;a$SCREEN$
  40 RANDOMIZE a: LET l=PEEK 236
70: LET h=PEEK 23671: POKE 65357
,l: POKE 65358,h: RANDOMIZE USR
65356
  50 PRINT "Data start for pic "
;(f);" = ";a'"(POKES : ";l,h;")"
: LET a=a+6912: REM replace 6912
as before
  60 NEXT f: POKE 65356,33: POKE
 65359,17
  70 STOP : SAVE "pics"CODE s,n*
6912+12
 100 DATA 17,0,0,33,0,64,1,0,27,
237,176,201: REM replace 27 with
24 for D_FILE only, 16 for top
two thirds, 8 for top third only
9999 REM Call with:- POKE 65358,
high byte of data start address:
 RANDOMIZE USR 65356
```

PROGRAM 2

```
   1 REM compactor
  10 INPUT "Title?",a$: IF a$=""
 THEN STOP
  15 INPUT "attribute value?",c:
 REM not needed for D_FILE only
  20 LOAD a$SCREEN$
1000 LET d=16384
1005 LET a=40000: REM data start
1010 IF d>22527 THEN GO TO 4000:
 REM  adjust 22527 if altering
machine code
1020 IF PEEK d=255 THEN GO TO 20
00
1025 IF PEEK d=0 THEN GO TO 3000
1030 POKE a,PEEK d: LET a=a+1: L
ET d=d+1: GO TO 1010
2000 POKE a,255: FOR f=1 TO 254:
 IF PEEK (d+f)<>255 OR d+f>22527
 THEN GO TO 2050
2010 NEXT f
2050 POKE (a+1),f: LET d=d+f: LE
T a=a+2: GO TO 1010
3000 POKE a,0: FOR f=1 TO 254: I
F PEEK (d+f)<>0 OR d+f>22527 THE
N GO TO 3050
3010 NEXT f
3050 POKE (a+1),f: LET d=d+f: LE
T a=a+2: GO TO 1010
4000 IF d<23296 THEN GO TO 5000:
 REM  not if omitting attributes
4005 PRINT a-40000: INPUT "Title
?",a$: SAVE a$CODE 40000,a-40000
: STOP : REM  40000=datastart
4009 REM remainder not needed if
omitting attributes
4010 IF PEEK d=c THEN GO TO 5000
4020 POKE a,PEEK d: LET a=a+1: L
ET d=d+1: GO TO 4000
5000 POKE a,c: FOR f=1 TO 254: I
F PEEK (d+f)<>c OR d+f>23295 THE
N GO TO 5050
5010 NEXT f
5050 POKE (a+1),f: LET d=d+f: LE
T a=a+2: GO TO 4000
```

Now you have your compacted codes separately on tape you need to save them as one long code length. Do this

the "attribute" prompt. If you haven't got the Table I supplied in the January column, you can calculate it yourself. Say the picture is drawn in blue ink on yellow paper the value would be 1 (ink) + 6 (paper) * 8=49. Now load in the SCREEN£ and wait. From now on the program takes over, and the compacting may take some time, so go and make a cup of coffee, or have a stroll around the garden. You've been hunched over that VDU for too long anyway! When the transfer is complete the number of bytes that the picture has been compacted into is displayed. Make a careful note of this, and the title you use to save the compacted code to tape. Repeat with a new attribute value and SCREEN£ until you have all you need, then just reply to the "attribute" prompt with ENTER.

Clyde Bish presents tips on full screen graphics.

Table B

```
        LD      HL, 16384
        LD      DE, datastart   ;POKEd before call
LOOP:   LD      A,(DE)
        CP      255             ;check for filled byte
        JR      Z, FILL
        CP      0               ;check for blank
        JR      Z, MISS
        LD      (HL),A
        INC HLHL
RET:    INC     DE
        LD      A,H
        CP      88              ;80 for 2/3 screen, 72 for 1/3 screen
        JR      C, LOOP
LOOP1:  LD      A,(DE)          ;replace with RET if D_FILE only
        CP      attribute       ;attributes start here. POKE attr value
        JR      Z, MISS1
        LD      (HL),A
        INC     HL
RET1:   INC     DE
        LD      A,H
        CP      91
        JR      C, LOOP1
        RET
MISS1:  INC     DE              ;routine to skip attributes not to
        LD      A(DE)           ;be altered
        PUSH    DE
        LD      D,0
        LD      E,A
        ADD     HL,DE
        POP     DE
        JR      RET1
MISS:   INC     DE              ;routine to skip D_FILE bytes not
        LD      A,(DE)          ;to be altered
        PUSH    DE
        LD      D,0
```

```
        LD    E,A
        ADD   HL,DE
        POP   DE
        JR    RET
FILL:   INC   DE              ;routine to fill bytes
        LD    A,(DE)
        LD    B,A
BACK:   LD    (HL),255
        INC HLHL
        DJNZ  BACK
        JR    RET
```

PROGRAM 3

```
1 REM Trader:from an idea by J.K.Moody
10 PRINT "TRADER": INPUT "Who is the Captain?",n$: CLS
37 LET s$="    ": LET t=0: LET b=500: LET y=5000: LET v=50: LET g=0: LET a=0: LET s=0: LET w=0: LET o=0: LET l=1: GO SUB 9900
95 GO SUB 9800
100 PRINT AT 0,10;n$''"Bank    $";b+y;s$'"Borrowed    $";y;s$'"Deck Space",v;s$'"Location",l$+s$'"General Cargo",g;s$'"Arms",a;s$'"Silk",s;s$'"Whisky",w;s$'"Opium",o;s$''"PRICES:"'"General  $";p(1);s$'"Arms       $";p(2);s$'"Silk       $";p(3);s$'"Whisky   $";p(4);s$'"Opium    $";p(5);s$
110 PRINT '"OPTIONS: 1.Buy 2.Sell 3.Sail"'("              4.Borrow" AND l=1)
120 INPUT "Choose your option ";c
170 GO TO c*1000
1000 INPUT "What do you want? ";p$
1070 LET p=(p(1) AND p$="G")+(p(2) AND p$="A")+(p(3) AND p$="S")+(p(4) AND p$="W")+(p(5) AND p$="O")
1075 PRINT #0;"You can afford ";INT ((b+y)/p): PAUSE 100: INPUT ;: INPUT "How many do you want? ";q
1155 LET v=v-q: LET b=b-(q*p)
1170 IF p$="G" THEN LET g=g+q
1180 IF p$="A" THEN LET a=a+q
1190 IF p$="S" THEN LET s=s+q
1200 IF p$="W" THEN LET w=w+q
1210 IF p$="O" THEN LET o=o+q
1250 GO TO 95
2010 INPUT "What do you want to sell? ";p$
2020 IF RND>.7 THEN PRINT #0;"Sorry, no buyers": PAUSE 100: INPUT ;: GO TO 100
2050 LET p=(p(1) AND p$="G")+(p(2) AND p$="A")+(p(3) AND p$="S")+(p(4) AND p$="W")+(p(5) AND p$="O")
2120 INPUT "How many to sell? ";q
2170 IF q>1 AND RND>.7 THEN LET q=q-INT (RND*q): PRINT #0;"You can only sell ";q: PAUSE 100: INPUT ;
2180 LET b=b+p*q: LET v=v+q: IF p$="G" THEN LET g=g-q
2210 IF p$="A" THEN LET a=a-q
2220 IF p$="S" THEN LET s=s-q
2230 IF p$="W" THEN LET w=w-q
2240 IF p$="O" THEN LET o=o-q
2245 IF b>100000 AND y=0 THEN GO TO 9000
2300 GO TO 95
3010 LET t=t+1: LET i=INT (y*.1): LET y=y+i: LET b=b-i
3020 INPUT AT 0,0;"Ports: 1.Hong Kong 2.Singapore"'"3.Macao 4.Bankok 5.Shanghai ";l
3050 GO SUB 9900: IF RND>.25 THEN GO TO 9700
3090 GO TO 95
4020 INPUT "Borrow or Repay(R)? ";c$
4040 IF c$="B" THEN GO TO 4500
4050 INPUT "How much? ";m
4080 LET y=y-m: LET b=b-m: GO TO 95
4500 INPUT "How much? ";m
4530 LET y=y+m: LET b=b+m: GO TO 95
9005 CLS : PRINT "You can retire with $100,100."''"You took ";t;" move";("s" AND t<>1): STOP
9710 PRINT #0;"STORM!": PAUSE 100: INPUT ;: IF RND<.5 THEN GO TO 9780
9740 PRINT #0;"Half your cargo is lost": PAUSE 100: INPUT ;: LET g=INT (g/2): LET a=INT (a/2): LET s=INT (s/2): LET w=INT (w/2): LET o=INT (o/2): LET v=50-(g+a+s+w+o): GO TO 95
9780 PRINT #0;"STORM OVER. ALL O.K.": PAUSE 100: INPUT ;: GO TO 95
9800 DIM p(5): LET p(1)=INT (RND*15+1): LET p(2)=10+INT (RND*14+5): LET p(3)=10*INT (RND*700+30): LET p(4)=100*INT (RND*26+5): LET p(5)=100*INT (RND*91+10): RETURN
9900 LET l$=("Hong Kong" AND l=1)+("Singapore" AND l=2)+("Macao" AND l=3)+("Bankok" AND l=4)+("Shanghai" AND l=5): RETURN
```

by adding up all the code lengths you noted, and adding 67. (This is for the machine code you'll need later to "uncompact" them). Let's call the answer T. CLEAR T-1, then load in the first compacted code to address T using

LOAD "title" CODE T

Load in subsequent compacted codes, adding the length of that code to the previous address, and noting the new load address. So if the first code was 2000 bytes long, the next would load in at T+2000, and so on.

Hopefully, when all the codes are in you'll have 67 bytes left below the start of the udgs. Load in the data from **Table A,** reading across each line, with:

FOR f=65301 TO 65367: INPUT i: POKE f,i: NEXT f

Now save the whole data and machine code block with SAVE "title" CODE T, 65368-T.

To use this compacted code in your programs you need to have a subroutine such as:

9999 RANDOMIZE a: POKE 65305, PEEK 23670: POKE 65306, PEEK 23671: POKE 65326, c: INK c-INT (c/8) *8: PAPER INT (c/8): RANDOMIZE USR 65301: RETURN

where variable a is the data start of the picture you want to call (noted when you made the one long code length), and c is the background attribute value used in the compactor program (49 in my earliest example).

Table B gives an annotated disassembly of the machine code so that readers who want smaller/two-tone/line drawings with no filled areas, (AND who understands what they are doing!) can alter the machine code to operate over less of the screen/ignore the attributes/ ignore 255s and so make it run faster. **(Program 2** will also need adaptation. Refer to the REMs).

Kingdom came

Next time we'll be looking at make strip-cartoon-type adventure graphics, "Redhawk"-style, but before I go I'll keep the promise I gave in an earlier article, and supply a simple program of the "Kingdom" kind for those who want something to work on to use as a driver for their graphics. The listing is given in **Program 3.** The purpose of the game is to accumulate £100,000 by astute, if somewhat shady trading practices. The scenario is the South China Seas, but could just as easily be smuggling along the Cornish coast, or whatever. Your ship can hold 50 units of cargo, the buying/selling price of which fluctuates. You start with £500 of your own, plus £5,000 you have borrowed and must ultimately pay back. Interest is added to this whenever you change ports. Oh yes. You may run into storms en route and lose part of your cargo.

The program, which will run as listed, is in a very simple format with a simple text screen display. As listed it takes up some 3K, but this could be shortened considerably using the byte-saving tricks I demonstrated earlier in this series. There are a few error checks. Writing these in is a good programming exercise. I leave it to your imagination to add the scenes using picsave, compactor, or any of the other techniques I've explained earlier.

One last bit of help, though, with PRINTs and INPUTs. Printing to the screen (with speech bubbles if you wish) is quite easy. Simply use

PRINT AT r, c; "text"

where r=the row, and c=the column you wish the text to appear.

For inputs you'll need to use a subroutine to simulate the normal input routine, but wherever you want on the main screen. Add **Program 4** to your main driver program, and set r and c to the row/column you want the input characters to appear, before you call the subroutine. Code 12 is delete (see p.183 of your manual) so CHR£ 8 (cursor left) is used to backspace before printing the replacement character. Code 13 in line 9995 is the code for ENTER, so the subroutine returns.

Now away to the pixel paper, and get sketching!

PROGRAM 4

```
10 REM input subroutine
11 REM r=row, c=column
9000 LET a$="": PRINT AT r,c;
9010 FOR d=1 TO 20: NEXT d
9020 IF INKEY$="" THEN GO TO 9020
9030 IF CODE INKEY$=12 THEN IF LEN a$>0 THEN PRINT CHR$ 8;" ";CHR$ 8;: LET a$=a$( TO (LEN a$-1)): GO TO 9010
9040 IF CODE INKEY$=12 THEN GO TO 9010
9050 IF CODE INKEY$=13 THEN RETURN
9060 LET i$=INKEY$: IF i$>="0" AND i$<="Z" THEN LET i$=CHR$ (CODE i$+32)
9070 LET a$=a$+i$: PRINT i$;: BEEP .1,12: GO TO 9010
```

Q-Liberator

David Nowotnik reviews the latest BASIC compiler for the QL.

Q-Liberator
Liberation Software
£69.95

Not long ago, there was only one BASIC compiler for the QL; now there are three. The first, Supercharge, was reviewed in ZX Computing a year ago. Next on the scene was Q-Liberator, closely followed (at least, in theory) by Turbo from Digital Precision.

Q-Liberator is supplied in an attractive video-cassette-sized case, with two microdrive cartridges and an 80 page manual. Of the two cartridges, one is a fully working 'copy', the other is a 'master'. The ingenious security protection system on Q-Liberator prevents users obtaining working copies of one of the essential compiler files by all means apart from use of the master. The user can obtain up to five working copies from this master, configured to any chosen storage device. Once a working copy is created, it can be used without hindrance from any other security system.

Using Q-Liberator is simplicity itself. The boot program loads the compiler's run time routines and SuperBASIC extensions used by the compiler. Once these are installed, the BASIC program to be compiled is loaded as normal, and the program compiled with the 'LIBERATE' command.

Q-Liberator compiles in two passes. The first pass produces a working copy of the BASIC program; the second pass produces the compiled code. The user can select the second passes to occur automatically after the first, if sufficient RAM is available.

There are several options available for the compiled code. In program development, Q-Liberator will produce a line number table and/or memory usage statistics, which help to identify errors. These options can be suppressed in producing the final version, to save space. The run time routines can be incorporated into the final code, producing a totally independent program, or they can be kept separate. The latter option can save a lot of space if you have several Q-Liberator compiled programs running concurrently.

Extended SuperBASIC commands can be incorporated into the Q-Liberator produced code. Up to eight machine code sets of new commands can be added, but you'll need to have a little knowledge of these routines (the start of the initialisation routine, and the start of the name table relative to the start of the code) to supply the compiler with the appropriate parameters. Other compiler options include setting heap, stack, and input buffer sizes, and switching on/off integer arithmetic modes. These options are available by inserting simple REM lines into the SuperBASIC program.

A number of SuperBASIC extensions are supplied with Q-Liberator, for use specifically with the compiler. Of particular interest are commands to turn on/off the cursor in the compiled program, and a useful set of error trapping commands which can replace the mixed bag of commands in the various ROM versions of the QL.

The producers of Q-Liberator make the claim for their product that it is fully compatible with SuperBASIC (bearing in mind that there are several commands e.g. LIST which make no sense in a compiled code). In reviewing this product, several programs were compiled. All compiled versions worked well, with little or no modification required to the BASIC programs to achieve successful compilation. The compiler picked up a missing END SELECT in one program, which QL's translator did not seem to mind. The friendly error and warning messages allowed easy interception of this and other minor problems.

The first thing to note is the difference memory size can make to operating speed; with more RAM, BASIC and compiled programs work faster. The test program compiled with Q-Liberator (with all default options set) produced significant savings on everything except the trigonometry test. This is not surprising, as trigonometry calculations require a lot of ROM time — whether in BASIC or as a compiled program. Inclusion of the run time module into the Q-Liberate compiled program makes no significant difference to operating speed.

The time saving on integer maths was disappointing. SuperBASIC doesn't perform true integer mathematics (integers are converted to floating point numbers before they are used), so a very large saving with use of variables with the '%' suffix on compilation was to be expected. It seems that Q-Liberator will not correctly interpret the '%' in a variable without the integer maths directive switched on.

Surprisingly, with the integer directive, the simple loop takes longer to perform, so care is needed with this directive to optimise operating speed. Integer mathematics with trigonometric functions is meaningless, but the compiler compiled and ran integer trig functions without complaint!

For comparison, the same routine was compiled with Supercharge, and the operating speeds are about the same: Supercharge has the edge on the simple loop and integer maths, Q-Liberator does slightly better with the FOR/NEXT loop and trigonometry test. Supercharge failed to compile RESULT procedure correctly (it provided an integer result), so that had to be altered to provide results for comparisong.

All file sizes were obtained using the INFO option in ICE. Surprisingly, all Q-Liberated files were larger than the corresponding SuperBASIC file (even without incorporating the run time module), and their size not predictable from the size of the original SuperBASIC file. Incorporation of the run time module adds another 8K to the size of the compiled program.

For the shortest (875K BASIC file) program, compilation took 180 seconds with microdrives and 115 seconds with RAM discs. In every case, loading the compiled program was much faster than the corresponding SuperBASIC program. This was particularly noticeable with the longer programs.

With its very close compatibility with SuperBASIC, and simplicity of use, Q-Liberator offers a distinct advantage over Supercharge. And for those who simply want their SuperBASIC programs to load and work faster, Q-Liberator has a lot to offer. Pity it is not priced a little more reasonably.

Liberation Software, 43 Clifton Road, Kingston upon Thames, Surrey.

MINDPLAY

Peter Sweasey is thawed out temporarily to review Silicon Dreams and the first GAC adventures.

Though the Ides of March may be approaching as you read this, the dungeon is currently several inches deep in snow. No £5 bonusses for me; I just have to content myself with making snow figure versions of Bryan, my barbaric editor, then jumping up and down on them.

It is so cold at the moment that my faithful Spectrum (this particular model is over four years old) has been affected. If left in its usual position overnight, half the keyboard behaves as if symbol shift is permanently being pressed; the remedy is to place the machine in the linen cupboard for ten minutes!

Several months ago I plugged a then fledgling adventure club called Adventure probe. Well, I'm glad to see the magazine has grown considerably since then, and you get a lot more for your £1 cover fee (or 75p if you take out a year's subscription). As Sandra Sharkey (the magazine's editor) says, reader feedback has improved it tremendously. There are various hints, reviews, and opinions; but what Probe seems especially strong on is help. Several pages are crammed full of readers' names, followed by lists of games that they are prepared to help with. Maps and solutions can be bought from Probe themselves for the princely sum of a first class stamp! Enclosed with each issue is Adventure Shop, which is a catalogue of amateur games for sale, some of which sound most intriguing. Something which particularly caught my eye was an ad for The Fix — an addition to The Quill which, though not written by Gilsoft, adds full sentence interpretation and a proper WAIT command. Generally, interaction between readers seems better than in most such clubs, so if you're an adventure freak, it could be worth buying a trial copy. Send £1 for a sample issue to: Adventure Probe, 73 Merton Road, Highfield, Wigan WN3 6AT.

A product called the Adventure Builder System also reached me this month, courtesy of CRL's awesomely feeble budget division, Alpha Omega. For just under a fiver, this may appear to offer a bargain Quill, but no such luck. It provides you with a set of routines and structures for you to write your game around — in BASIC! The coded routines are much faster than BASIC would be, speed supposedly being that language's main disadvantage. This really is a pointless release. GAC, the Quill and Gilsoft's soon-to-be-released Professional Adventure Writer all enable you to write much faster and more complicated adventures, with almost the same flexibility as BASIC but less of the hassle. BASIC programming is time-consuming and wasteful of memory — that's why Quill-type utilities were designed in the first place. Besides, ABS is only £4 cheaper than the recently re-priced Quill. If people want to write for pleasure, I think they will pay the extra; if they want to write commercially, they'll have to.



HELPLINE

I've been having a little new-year cleanout down here, and I've discovered something of a backlog of old problems to be dealt with. Sincere apologies to anyone who has had to wait for a reply: I should have cleared the system by the time you read this, so you will not have to wait so long in future.

P. Logue from Abingdon asks how to pay tribute to Nero in **Roman Eureka**. Anthony Dunn kindly supplies the answer: simply HAIL him. Another quickie from Mick Hitchman, who needs the words to climb out of the lift in Interceptor's **Aftershock**, once he has removed the ceiling panel. Er, CLIMB OUT OF LIFT...

David Brown of Standord-Le-Hope would like to enter the Ivory Tower in Ocean's **Never Ending Story**. As far as I know, it's impossible in Part One — you certainly do not need to. At the start of Part Three you'll find ourself outside it — UNLOCK DOOR with the gold key from part two (found west from the torture chamber). Thanks to Charles James for a Never Ending solution.

Peter Davies — who describes himself, quite accurately, as "the hermit who is still playing **Planet Of Death**" — wants to know how to get the mirror without breaking it. From the cavern its housed in, go to the adjoining wind tunnel, and take and wear the gloves you'll find there. With these, pick up the man and drop him. As a result, he'll be removed from the mirror, so you can shoot him without incurring bad luck.

Some **Hobbit** queries next. Someone called "Vandepeoltom" from Belgium wants to find the magic ring in the dark, stuffy passages. From the Goblin's dungeon, go west out the window, then SW, D, WAIT for the goblin to appear. N, SE and E. Les Acton from the Isle of Wight has broken through the spiders' web to the Elves' Clearing. "How do I go through the magic door?" You need to wear the ring, then EXAMINE DOOR. Keep WAITing and wearing the ring alternatively; eventually the door will open as an elf sweeps by. Go NE immediately this happens, before the door closes again. Norwegian Sam MacDonald wishes to kill the dragon without anyone's help. I think this is impossible: you need to pick up Bard at Lake Town (literally! — CARRY BARD). Then head to the dragon's treasure hall, avoiding him on your way. If he's there when you arrive then SAY TO BARD "SHOOT DRAGON". If he's not, go east, wait for a few moves (wearing the ring might help), then east again and he should be there.

Adam Cochrane needs help in **Denis Through The Drinking Glass**. The important thing is not to go out through the window to start with. Dress, take the letter, radiation suit and Cherry Blossom, move the sheets from Maggie's Den to your own, take and wear the teeth, then out into Downing Street and BUY TICKETS.

K. Thain wants to board a train in Double Play's **The Moreby Jewels**. I've never played this, so I'm indebted to the help of Cheshire's John Hunter. In the garden of your house, GET ENVELOPE (it's sticking out your door), and OPEN it. Go to the railway station, GET and READ TIMETABLE, GO TOILET, USE TOILET, U, N, SHOW TICKET, ENTER PLATFORM and BOARD TRAIN. "K" is also stuck in K-Tel's **Castle Colditz**, which again I have not played, but which again John Hunter helps with. USE GRENADE in the passage of rats. Avoid the radio room, you don't need to visit it to complete the game. John also helped a reader out with Runes Of Zendos (the solution is too complicated to print here) so many thanks indeed.

HELP!
Anybody out there solved Vera Cruz or Questprobe 3? Please send me a solution if you have — I'll be happy to help with another game in return.

Wave goodbye to adventuring blues ...

with new system, Helpline automatic. No additives or preservatives! And now at special sale prices — every solution must go! If you have a nasty Boggit problem, troubles with vampires in Dracula, or are unable to ring that bell in Hunchback — well Mindplay (TM) can help.

Yes, your super, soaraway ZXCM has devised this simple system — even the hamster could use it!

Step 1: Take a pen.
Step 2: Fill in the coupon.
Step 3: Send it off to the address below.

Follow those and before you can say Sid, your posers will have been solved. Please read the small print now...

British correspondents, please enclose a stamped, addressed envelope if you want a personal reply rather than wait some months for the magazine to come out. If you are writing from abroad, just enclose an envelope — I'll add the postage. I try to respond within two months but I can take longer (on the other hand you might receive an immediate reply). I ONLY DEAL WITH ADVENTURES. Not arcade games; or technical problems (write to Crosswires about those); or arcade adventures (Gargoyle games included, not even Heavy on The Magick). Please put the name of the game you're writing about on back of the envelope. And PLEASE don't write letters asking for general solutions — I just don't have the time to write these out for everyone. (They can be obtained, however, from clubs like Adventure Probe — see this month's introduction).

The address is: Mindplay, ZX Computing, No 1 Golden Square, London W1R 3AB. How many other magazines solve your problems like this AND leave your hands silky smooth?

Title: ....

Company: ....

Problem: ....

....

....

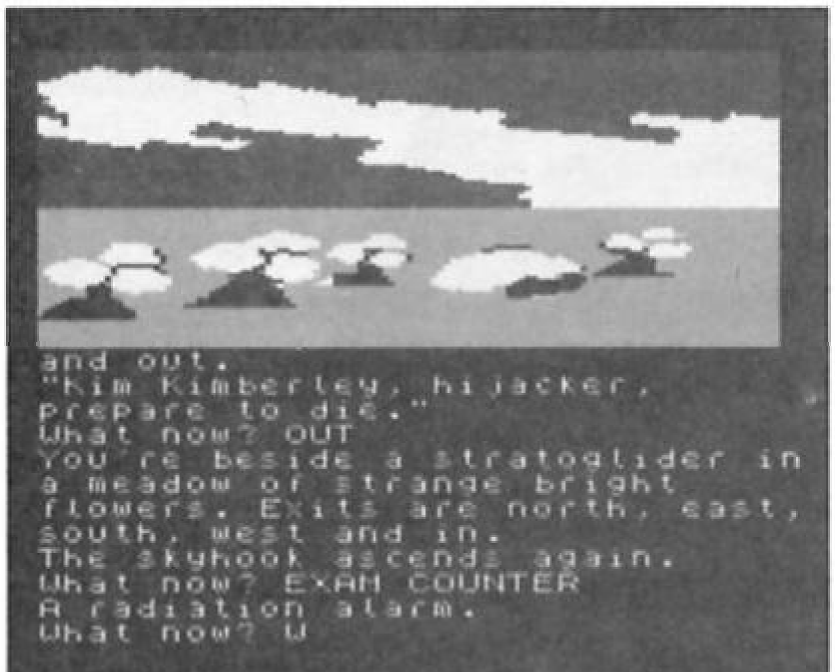
I can help solve: ....

Name: ....

Address: ....

....

SILICON DREAMS
Rainbird
£14.95



Following "Jewels of Darkness" last year, this Rainbird offering is another collection of three revamped, repackaged Level 3 adventures. This time the games are in the science-fiction genre, and tell the extraordinary tale of Kim Kimberly. In Snowball, you play Kim and, re-awoken from deep-freeze storage, must save the eponymous spacecraft from crashing into a star. Return to Eden finds you crash landing on the planet of the same name, having been wrongly accused of sabotaging the Snowball. Fight through the alien jungle, then survive the deserted future city of Enoch, built many years ago by advanced robots preparing for the colonists' arrival. It's now in a precarious state and staffed by droids hostile to your intrusion. Worm In Paradise, set a century later, sees you as a citizen of Enoch, an extreme right-wing state run by robots, in which all must conform and Kim is worshipped, Lenin-style. Challenge the system, and save the world!

I searched the box carefully on receipt — surely Telecom haven't finally listened to all the criticisms? Yes they have — to the unending gratitude of adventurers, the dreaded Lenslok has been scrapped! Otherwise, I experienced a powerful feeling of deja vu when reviewing Silicon Dreams. It contains almost exactly the same unnecessary faults as Jewels Of Darkness, as well as that game's good points.

All three adventures are large, and full of ingenious puzzles. One of the games alone would keep most people occupied for months. Vocabulary is large and complex sentences are allowed. A highly readable instruction booklet also includes a novella — 'Eden Song', which far from being some overblown epic laden with silly names is unexpectedly humorous, if not entirely in the style of the games.

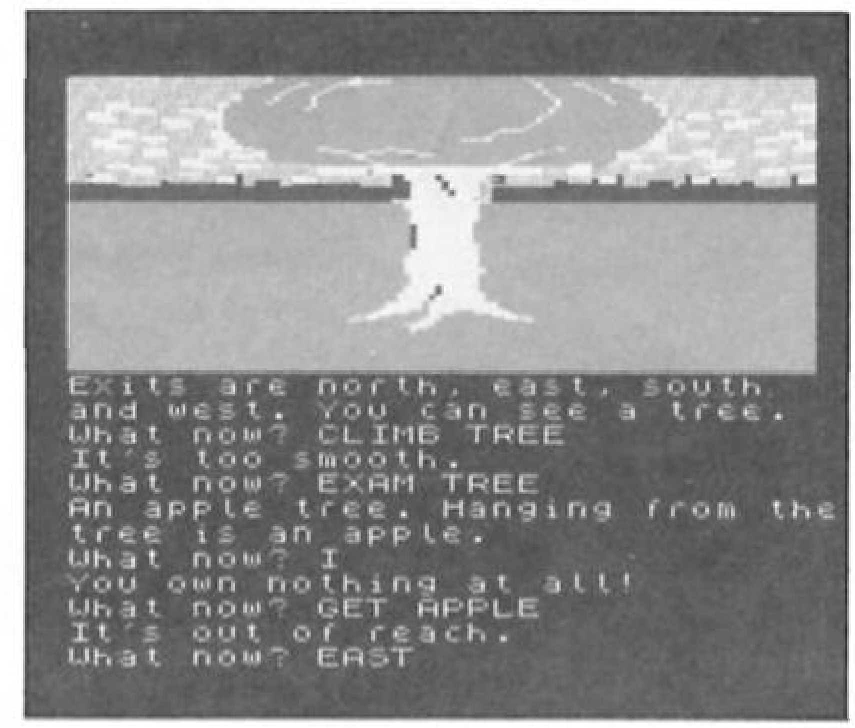
But the potential for a masterpiece has again been wasted through silly errors; like painting the smile on the Mona Lisa with a blue crayon. First up, my traditional moan: the graphics are truly abysmal. Blotchy, often unrecognisable and, in this collection, so simple in design as to render them even more pointless than in the past. The screenshots here do not do them justice — you have to be close to the screen for the full impact. In the case of Eden, the graphics are worse than those included on its original release! I always thought the Spectrum's attribute problems were to blame for the pics' dreadfulness, but the Atari ST screenshots shown on the packaging are almost as useless. Why Rainbird allow this sloppiness in what is supposed to be the cream of British adventures is beyond me: though at least I see ST users will be getting digitised Geoffrey Dawson graphics on the next joint release. Fortunately, expanded, text-only versions of each adventure are provided. The quality of the text is so much better in these versions that they feel like different adventures. Certainly they are much more atmospheric and they are easier, since the strange technology is explained better.

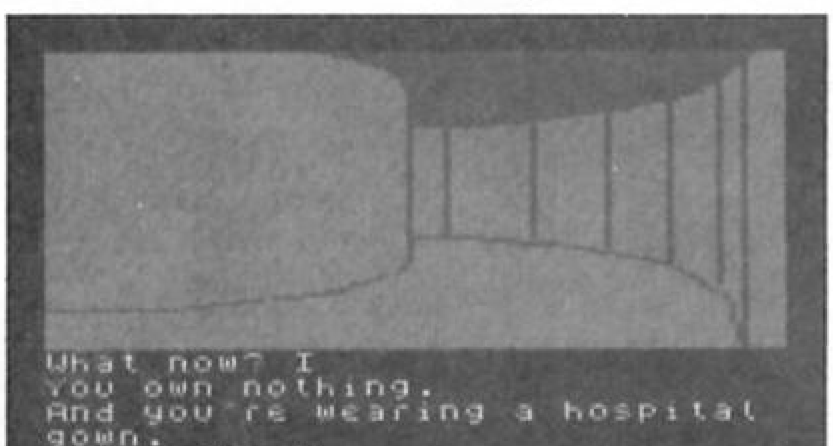
Presentation is poor: all text is printed in yellow on black in the normal Spectrum character set, which makes an untidy mess. Why no colour, or better spacing and a more readable character set? Still no loading screen either. Instead of an atmospheric picture to set the mood we have "Level 9" printed in several different colours — this looked crude even when I first encountered it several years ago.

RAMSAVE/LOAD is standard on Quilled games, so its omission from a major release is disappointing. At least it could have been included in the text only version, using some of the memory freed by removing the graphics. Its absence wouldn't be so bad if conventional tape loading was not so drawn out. The security device — giving you a page and line reference from Eden Song and asking you which word it refers to — is not as irksome as Lenslok; however, you have to do this every time you RESTORE from tape, which for most adventureers I suspect will be often, as death lurks round every corner. Why not just ask for a word at certain key points in the game?

As with JOD, I must express how disappointed I am that the potential of these games has not been realised. I sincerely hope Level 9 will take notice of some of the criticisms — they are meant to be constructive. That said, the package offers improved versions of three excellent adventures at half the original cost. Individually they would have been Monster Hits — so together they are, too.



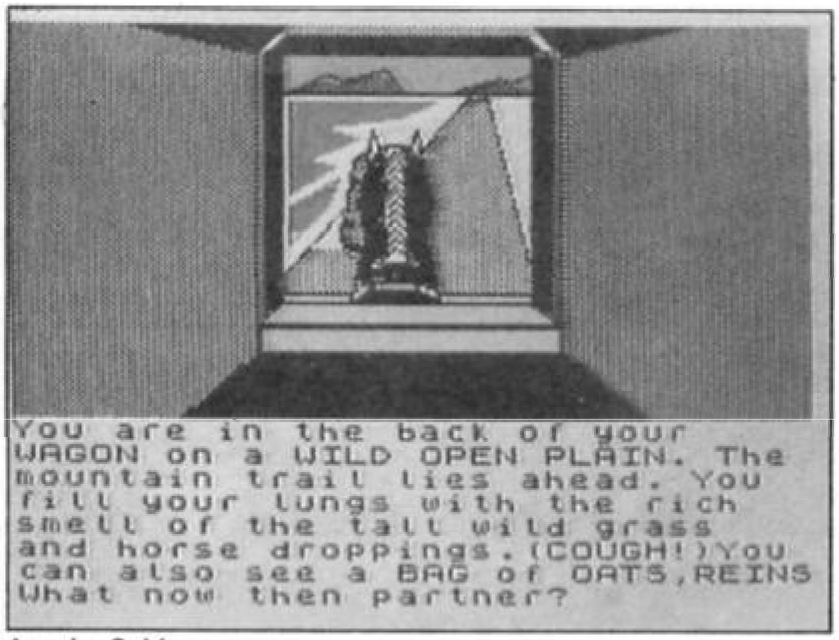




WINTER WONDERLAND/ THE LEGEND OF APACHE GOLD
Medallion Graphic Adventures (Incentive)
£7.95 each

These two adventures are reviewed together because their faults are similar; and they're the first I've seen written with Incentive's Graphic Adventure Creator, so are they better than your average Quilled piece?

Apache Gold sees you as cowboy Luke Warm (groan), in search of the treasure buried with an Indian Chief. The writer



Apache Gold

is Peter Torrance, described as a "fantasy author" on the packaging; in fact a credit arising from his previous successes, Firebird cheapies Subsunk and Seabase Delta. Winter Wonderland casts you as an anthropologist — original at least — who must investigate a Tibetan civilisation, which has never been in contact with the rest of mankind, yet developed at the same rate.

Immediately you notice the graphics — appropriately so considering the utility's name — which are faster, bolder and much bigger than in Quilled games, and remain on screen rather than scroll up. The pics are jolly enough in Apache Gold, but poor and repetitive in Winter Wonderland. In neither game are they consistently up to the design standard of The Hobbit from four years ago! The text window is too small, with no border separating it from the graphics. Because everything scrolls out of sight so soon, you have to keep LOOKing, but this causes the picture to draw again on top of itself, with a resulting, unnecessary delay. Text display is black on white (or on blue, deeper into Winter Wonderland), with no redesigned character set. Combined with slightly sluggish printing times, this means that the games don't appear anything like as slick as, for example, The (Quilled) Colour Of Magic.

An advantage of GACed games over Quilled ones is the editing facilities: you can enter commands in upper and lower case, punctuate, and use cursor keys. Also, these games sometimes require sentences with more than two key words. However, despite their technical superiority, the actual adventures are fairly primitive by today's standards. Description in both is below par, dull in Winter Wonderland and simple, almost childish, in Apache Gold. Torrance STILL has an ANNOYING habit of breaking into CAPITALS for no APPARENT reason, and PUNCTUATION problems. Neither game is particularly atmospheric or innovative — just fairly traditional puzzle solving, marred in Winter Wonderland by "instant death locations", a dumb idea which should have died out aeons ago. Torrance gives clues which are laughable for their lack of subtlety. Both games have limited vocabularies and a disappointing lack of responses. And the instructions are too skimpy — nothing on tape storage, or (in Winter Wonderland) on how to communicate with other characters.

In short, what we have here are two unremarkable adventures, which could have been written a year or two ago. They would make ideal, indeed particularly entertaining, budget games. However, eight pounds is a ridiculous price for such mediocrity, so steer clear. What I find surprising is that these were the best of the many games Incentive were sent (Medallion is Incentive's own label for such games). Evidently there are no hordes of potential McNeills and Austins out there — unless you can prove otherwise...



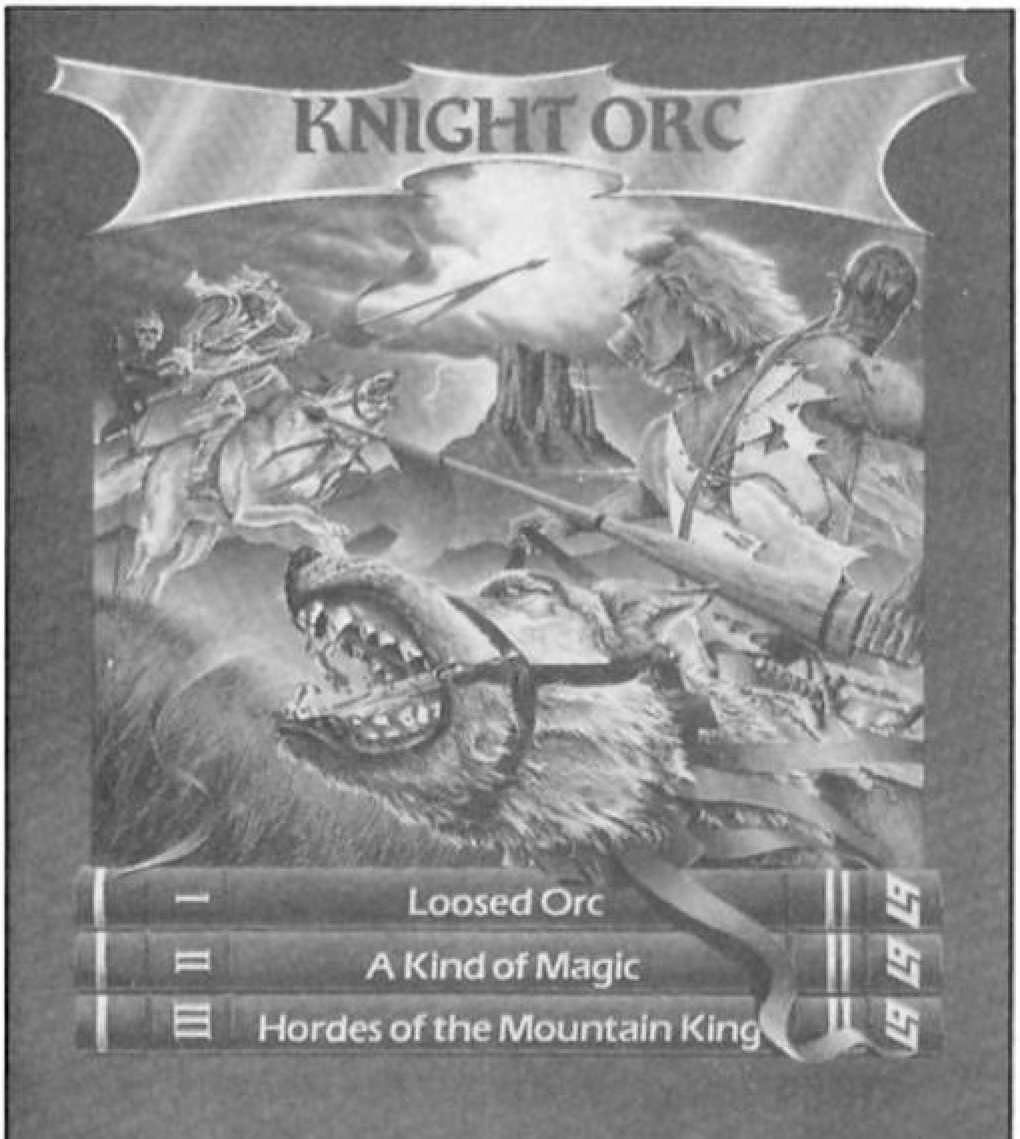
GRIM

RAINBIRD

Rainbird are probably the only software house never to get a bad review from ZX. So what's their secret?

When I went along to visit Rainbird Clare Edgeley had just tidied up her desk to keep the safety officer happy — "my desk had everything wrong with it".

Depressingly, Clare's bit of floorspace looked like an oasis compared to the heap that passes for my own desk. In fact Rainbird's offices, in a modern office block in the West End of London, seemed a bit cheerier and more well organised than many of the software houses I've visited in the past. But then, whereas most UK software houses began as back-bedroom outfits and slowly expanded into small companies as the piles of paperwork and machinery got out of hand, Rainbird came at the industry from the opposite direction. As part of the Telecomsoft group of companies (which also includes Firebird, Beyond and Odin), Rainbird are actually a small offshoot of the vast British Telecom organisation. But like most software houses they've managed to avoid too much office stuffiness and remain fairly informal.



Clare is Rainbird's P.R. Manager but she is also an ex computer journalist, having worked for Computer and Video Games and an obscure competitor called Sinclair User.

Former software reviewers moving into PR face a potential quandary. Do they wholeheartedly push a product they would have panned in their former role? Fortunately that problem doesn't arise with Rainbird since I can't remember any of their products ever getting a bad review. Unlike many companies that make similar claims, Rainbird have genuinely built up a reputation for releasing a small number of high quality products.

This is probably due to the fact that the label was set up to be the 'quality' part of the Telecom group.

"We're the top end of the range — high quality", Clare told me. "We'll look at anything. Some of the best games have landed on software houses' doorsteps as complete games, but if someone comes to us with a scenario or an idea that we like, we'll develop it. But we wouldn't bring out just anything — if we're offered something that's not worth our £14-£17 price range we won't accept it."

But isn't it difficult to continually maintain a lead when you're dealing with a machine like the Spectrum which has been taken about as



far as it can go by now?

'Yes, but converting down from big machines helps. We produce conversions for about ten machines and some of our games start on the Amiga and ST and are converted down onto the Spectrum so we sometimes manage to do things that we didn't think could be done on the Spectrum. Starglider 128 has digitised speech which we didn't think we'd be able to do. I don't think there's a single bit of memory left in there."

An obvious example of this 'converting downwards' policy is Art Studio, the graphics utility that brought to the Spectrum a system of icons and menu-driven controls formerly only seen on much more powerful business machines. Mind you, it's not always a case of converting down onto the poor little Spectrum. Rainbird's collaborations with Level 9 (the Jewels of Darkness and Silicon Dreams trilogies) started on the Spectrum and have moved onwards and upwards through a whole range of machines and are doing very well in both America and Europe.



The Knight Orc Cometh

The last two Level 9 products from Rainbird were upgraded versions of earlier releases, but coming soon is Knight Orc, a brand new L9 game in three parts, and something of a departure from their old style. In it you play an Orc setting out to gain revenge on the humans who spend all their time victimising Orcs in fantasy novels and games (there's a twist in there though, but I'm not going to tell you about it). The game features a number of independent characters that you'll have to deal with, and whose help you'll need if you're going to complete the game so you can't just go around KILLing everything in sight.

After making the coffee, Clare shuttled me into the computer room where I met Gary (also ex C+VG) and Paul ("our resident wiz — he does manuals and things"). The Spectrum version of the game wasn't yet complete, but Paul ("*he's* the resident wiz, I'm the *arch*-wiz") booted up the graphics for the Amiga version to show me.



Pete Sweasey, our resident adventure fiend, will be pleased to hear that L9 have adopted a totally new style for their game graphics in Knight Orc. They've commissioned a number of pieces of original artwork and *digitised* them. The result, on the Amiga anyway, is pretty impressive, with some pictures that look almost impressionistic, and others that have almost photographic clarity of detail (fingers crossed they won't look too bad on the Spectrum).

Knight Orc is due out for the Spectrum shortly, as is the 128 version of Magnetic Scrolls' adventure, The Pawn. In the meantime though I decided that I should get back and tidy up my own desk in case the safety officer dropped by.

GAUNTLET™

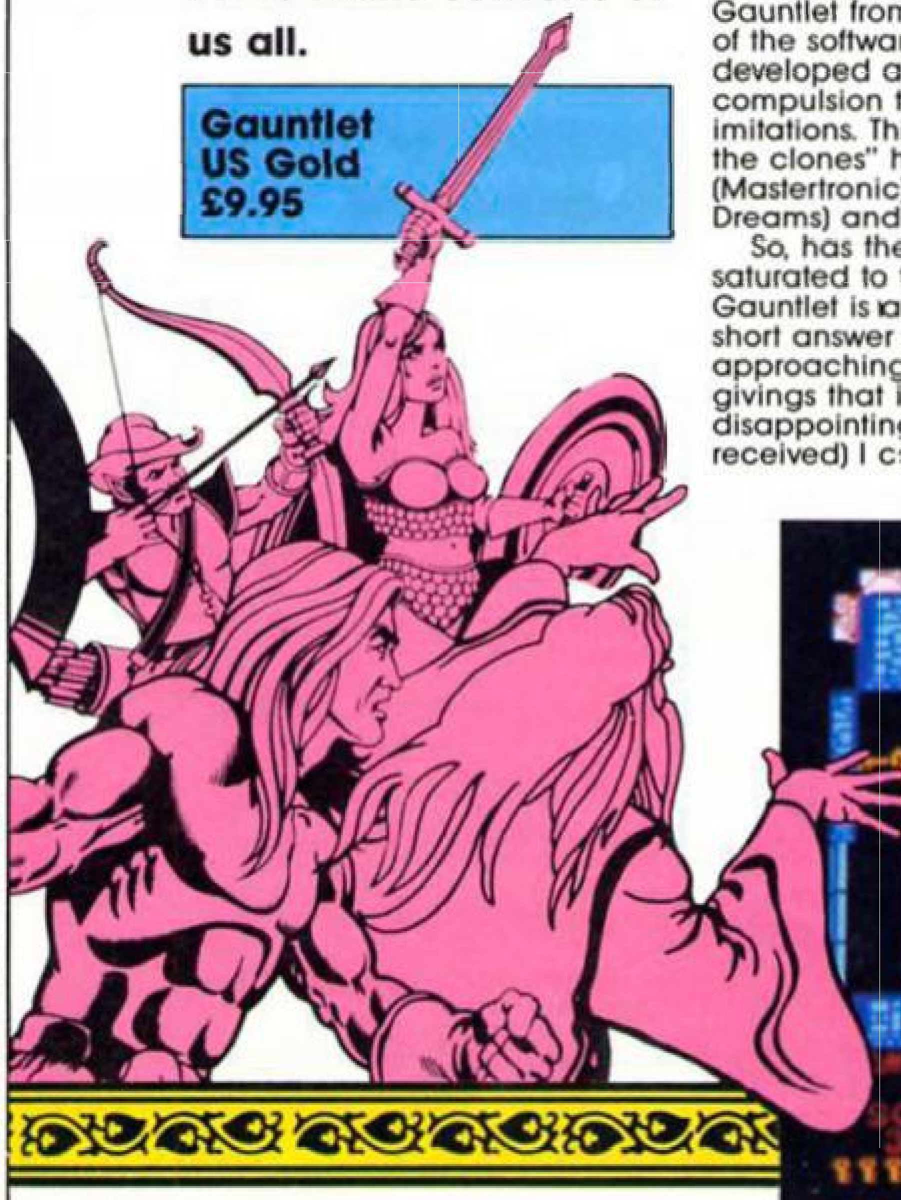






The arcade conversion set to make converts of us all.

Gauntlet
US Gold
£9.95



This is the game that everyone has been waiting for. During a wait for the official version of Gauntlet from US Gold, the rest of the software industry has developed an overwhelming compulsion to produce Gauntlet imitations. This chorus of "send in the clones" has brought us Storm (Mastertronic), Dandy (Electric Dreams) and Druid (Firebird).

So, has the market been saturated to the extent that Gauntlet is an anti-climax? The short answer is no. Despite approaching the game with misgivings that it could be disappointing (given the hype its received) I can report that the official Gauntlet is by far the most enjoyable and addictive of them all.

The original arcade game could be played by up to four players. On the Spectrum of course, this is whittled down to two (thankfully both can play with a joystick) but it is the option of cut and slash teamwork as you clear dungeon after dungeon of nasties that adds a whole new dimension to playing a Spectrum game. You would expect this two player option to slow down the action but surprisingly there is no loss of speed and the action is frenetically fast.





THOR the Warrior





THYRA the Valkyrie

You have a choice of four characters and, if playing the two player game, it's best to choose characters that have complimentary characteristics. Thor, for instance, is a warrior with excellent muscle power and good in close combat but a dunce when it comes to casting spells. You might want to team him with Merlin who gets pummelled badly in hand to hand fighting, but can use his magic power to wipe out monsters and the generators that spawn them.

Thyra the Valkyrie's speciality is hand to hand combat and Questor the Elf relies on his magic charms to make up for moderate abilities at fighting.

As for adversaries, the dungeons are overflowing with them, ranging from ghosts to Death, whose fatal charms will drain you of health unless you have the spell to kill him. In addition there are Grunts, Demons, Lobbers and Sorcerors all with their own characteristics and each requiring a slightly different approach to despatch them most effectively.

Be warned that due to the complexity of Gauntlet (512 levels) there is a lot of time spent rewinding and loading from tape. If you don't find Gauntlet compulsive this could be a drag but if you become addicted these breaks can become a welcome respite from the unrelenting battle.

Dungeon plays

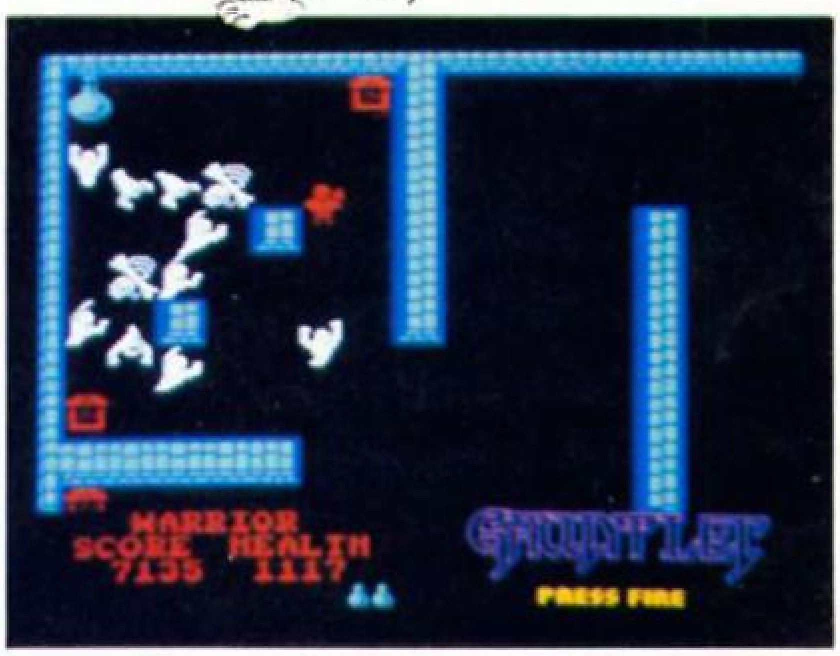
The aim of the game of course is to fight, barge, charm, shoot and punch your way through those 500-plus levels. It's not a game for dedicated strategists. For one thing the swarms of dungeon dwellers threatening you leave little time for planning ahead. There are, however, ploys you'll need to implement if you want to survive. Unless you are an advanced player you can forget about the points tally — worry instead about your health score. Go for all the food you can pick up (some of it is poisoned but after a while you can spot the difference) initially I found myself shooting food by mistake in the heat of the fray (not a good idea).

If cornered by monsters try to destroy the generators, you can't clear a screen completely until you have. Mass slaying of roomfuls of magic ghouls may make you feel good but think of your health which will be sapped by each individual contact.

Gauntlet holds a lot of surprises; the graphics are vivid and maintain their clarity, even on a screen teeming with detail. The speed of the game will satisfy even the hardened arcade player, the lifespan of the game will keep the value-conscious happy and there are enough elements to it to satisfy those who want something more than to just shoot things. Of course if you just like shooting things and know somebody else who likes shooting things too, then this game will seem very close to heaven.

MONSTER HIT





THE DISCOVERY COLUMN



Advice for Discovery Disc owners from John Wase

I suppose I've asked for it, but over the last few months quite a few enquiries have been made about the capacity of Discovery, principally, I suppose, because I mentioned that it is possible to add a Double Sided Double Density (DSDD) 5.25" drive. So, to get things straight, let's look at just what there is inside that black metal box.

The Drive

As fitted to all standard Discovery units, this is a 40 track single sided three and a quarter inch drive, known in the computer trade as a 'quarter megabyte drive'. The tracks are recorded as concentric circles. The mechanism reads 40 tracks on just the one side of the disc. So any old three and a half inch disc which has *at least* 40 tracks and which is the right shape to fit the mechanism will do. If you insert an 80 track disc (that is merely one in which the oxide particles are so small and the thing so precise that it is possible to put tracks much closer together than the 40 track disc without there being interference between one track and the next) the mechanism is only able to record 40 tracks on it.

There is a power supply unit included on Discovery's circuit board which powers both Discovery and Spectrum. The power supply which came with early Spectrums had so little power to spare that there was not always enough to work Opus's early Spectrados systems, so in some cases these would not work: the Discovery was a refinement of this system and its PSU now has plenty of power to spare.

The interface contains a read only memory (ROM) which intercepts error messages and is paged in on receiving one of its own commands. If you look at the Spectrum's memory map, you will see that the Spectrum's ROM occupies the first 16K. Discovery ROM occupies the first 8K, and is switched in and out alternately with the Spectrum ROM. As you know, the Spectrum ROM can be PEEKed to see what's in each address, but if you try to POKE a value into an address, it takes no notice. Similarly, the discovery ROM contains unalterable information. This method of operation is exactly like interface 1: however, discs are operated instead of microdrives, and, as the code is written rather differently, the hook codes which operate from machine code are different. So the two are compatible from any BASIC program, but not in respect of machine code.

Formatting

One of my correspondents recently was very uptight because the capacity of a quarter megabyte drive is only 178K. This, he growled, is grossly misleading. And so it is, but there is a good reason. Information recorded on a casssette tape is recorded sequentially: you have to grind through the lot until you come to what you want. A disc immediately accesses your chosen program. This is because each track on the disc is further divided up by the operating system into sectors or blocks, according to a set plan dictated by the software (the firmware on the Discovery ROM), the Directory or Catalogue of the disc. This directs the drive to read so many sectors, or, alternatively, to save to so many sectors at particular locations. The number of sectors used is arrived at as a compromise. Imagine a disc divided up into only one sector, and imagine trying to save 1K programs on it. As you can't direct the thing to start half way through a sector, you could only save one program, which you could access instantly. Moreover, I guess that you could make this program almost 250K long by a bit of fiddling about. Imagine dividing the thing up into one byte sectors. You would need a catalogue miles long for the thing to read, and whilst economical up to a point, access times would be enormous. So the choice of sector length, etc., is a compromise between speed and waste. Discovery is pretty reliable, has reasonable access speeds, and wastes relatively little of the disc.

Second disc

When a second disc is added, the drive mechanism printed circuit board has provision to enable one to configure it as disc 2, and a Random Access Memory (RAM) chip has to be added inside Discovery. This is because the default values in the Discovery ROM all refer to Disc 1, 40 track, single side. The RAM chip occupies the memory locations immediately above the Discovery ROM and is paged in with it: on sensing the presence of the RAM chip, the ROM empties its default values into the RAM chip. RAM, unlike ROM can be altered: it can be POKed as well as PEEKed. If you read n a DSDD disc in drive 2, assuming that that is the mechanism which you have plugged in there, then the firmware senses that the drive is not 40 track and single sided, and the default values for drive 2 are adjusted in the RAM to match those of the disc read in. Before you can use such a system you must have a disc to read in or a program to configure it, hence the frustration

of some who have fitted extra capacity drives but have only been able to get them to FORMAT to 178K: the values are set at default until something tells the system differently. Finally, if you have only the standard drive, fitting the RAM chip is not really going to help you very much. True, you will be able to adjust the sector length and one or two other things, but this will never enable you to store a megabyte, as some of my more optimistic correspondents had hoped. I hope this will help to clear up some of the misconceptions.

I mentioned only a couple of months ago that I had seen nothing which used Discovery's random access facility, so Jim Burke of Barnet, Herts, promptly sent me DISCLIST3. Although this program lacks some polish and finesse, as Jim would be the first to admit, it nevertheless does the trick, and that is the object of the exercise. DISCLIST3 keeps a list of all the program, and associated datafiles, printfiles and so, which are kept in random access file called "discs3". Once you have this, you can select any item by title, by the type of item or by the disc that it's on. The program is listed in Figure1. It's pretty self explanatory, although I found that I had to put in an extra line – 7 OPEN 3;"t" to get it to print. The REMs in lines 10 and 145 also need to be removed: they are there to remind you that certain parameters need some alteration to enable much longer lists to be accommodated.

The Program

The first time that you RUN this program, you will have to do GO TO 20 when the program can't find the random access data file "discs3" asked for in line 15: not surprising as you haven't made one yet. Lines 45,50 and 55 line up the list on the screen rather ingeniously. I printed it out (Figure 2) so that you can see what this data file looks like. Each entry is given a number, on the left. Then follows the name of the entry. Next a letter indicating the type of item. Finally, a number for the disc on

```
   5 REM "DISCLIST3"
   J.N.Burke 1986
  10 REM DIM n$(200,15)(or 19?)
  15 LOAD *1;"discs3" DATA n$()
  20 LET r=150: REM  Last entry
for checking*********************
  25 OPEN #4;"m";1;"discs2"RND15
,200: CLS
  30 PRINT AT 0,12;"OPTIONS";AT
2,0;"0 List Entries.";AT 5,0;"1
Revisions, Additions or Fresh St
art.";AT 9,0;"2 Pick an Entry By
 Number.";AT 13,0;"3 Select by T
itle, Category or  Disc Number."
;AT 18,0;"4 To SAVE the program
and File.";AT 20,0;"5 To STOP."
  35 INPUT "Number ? ";c$
  40 IF c$<>"0" THEN GO TO 65
  45 IF c$="0" THEN CLS : FOR g=
1 TO r: IF g<=9 THEN LET a$="  =
 ": PRINT #2;g;a$;n$(g)( TO 15)
  50 IF g>=10 AND g<=99 THEN LET
 a$=" = ": PRINT #2;g;a$;n$(g)(
TO 15)
  55 IF g>=100 THEN LET a$="= ":
 PRINT #2;g;a$;n$(g)( TO 15)
  60 NEXT g: GO TO 370: GO TO 30
  65 IF c$="1" THEN CLS : GO TO
100
  70 IF c$="2" THEN CLS : GO TO
145
  75 IF c$="3" THEN CLS : GO TO
190
  80 IF c$="4" THEN CLS : PRINT
AT 10,12;"SAVING": GO TO 380
  85 IF c$="5" THEN GO TO 95
  90 IF c$<>"1" OR c$<>"2" OR c$
<>"3" OR c$<>"4" OR c$<>"5" THEN
 BEEP .1,20: GO TO 35
  95 CLOSE #4: STOP
 100 PRINT "NOTE:-"'"DO NOT USE
""RUN""unless the  DATA has been
 SAVED"''"   Revise Lines (20),4
5,110,255,280 and GOTO 105 for R
evisions, Additions or for a Fre
sh Start."''"(GO TO 30 and Selec
t 2 to pick a seleted  number)."
''"(GO TO 30 and Select 3 to Sel
ect by Title,Category or Disc)."
''"(GO TO 30 and select 4 to SAV
E).": CLOSE #4: STOP
 105 CLS : PRINT "Title (max 10)
  (ZZZ to STOP)."''"TYPE A-rcade
, S-tatic, G-raphic U-tility, P-
uzzle, p-attern,    F-iling) or
M-isc.  DISC Number"
 110 FOR g=139 TO r: REM From Nu
mber that is "ZZZ" TO 150 etc***
*******
 115 INPUT "TITLE(<=10)? ";n$(g)
(1 TO 11): IF n$(g,1 TO 3)="ZZZ"
 THEN GO TO 140
 120 INPUT "TYPE ?(A,S,G,U,P,p,F
,M) ";n$(g)(12 TO 13)
 125 INPUT "Disc ";n$(g)(14 TO )
 130 PRINT #2;g;"=";n$(g)( TO 15
)
 135 NEXT g: REM g=136 26/11/86
 140 PRINT FLASH 1;''"Now SAVE (
GO TO 350)": CLOSE #4: STOP
 145 CLS : REM OPEN #4;"m";1;"di
scs3"RND15
 150 PRINT "Input ""999"" to Sto
p"
 155 INPUT "which record?";g
 160 POINT #4;g
 165 PRINT "No. ";g;"=";n$(g)(1
TO )
 170 INPUT "Next-";g
 175 IF g=999 THEN GO TO 185
 180 IF g<>999 THEN GO TO 160
 185 CLOSE #4: STOP
 190 REM OPEN #4;"m";1;"discs3"R
ND15
 195 CLS : POKE 23658,8: INPUT "
Title(T),Category(C) or Disc(D)"
;k$
 200 IF k$="T" THEN GO TO 220
 205 IF k$="C" THEN GO TO 225
 210 IF k$="D" THEN GO TO 240
 215 IF k$<>"T" OR k$<>"C" OR k$
<>"D" THEN BEEP .1,20: GO TO 195
 220 POKE 23658,0: CLS : INPUT "
Title(NB Case)? "; LINE t$: GO T
O 205
 225 CLS : PRINT #2;AT 10,10;"Ca
tegories:-";AT 13,0;"A(rcade).G(
raphic).S(tatic game)p(atterns).
P(uzzles).F(iling).  M(iscellane
ous).U(tility).": POKE 23658,0:
INPUT "Category(NB Case)? "; LIN
E c$
 230 INPUT ,"LPRINT?(y/n)";w$: I
F w$="n" THEN LET q=0: GO TO 280
 235 IF w$="y" THEN LET q=1: LPR
INT "      CATEGORY ";c$: GO TO 2
80
 240 POKE 23658,0: CLS : INPUT "
Disc? ";d$
 245 POKE 23658,0: INPUT ,"LPRIN
T?(y/n)";v$: IF v$="n" THEN LET
p=0: GO TO 305
 250 IF v$="y" THEN LET p=1: LPR
INT "      DISC ";d$: GO TO 305
 255 FOR g=1 TO r: REM **********
 260 IF n$(g)(1 TO LEN t$)=t$ AN
D p=0 THEN GO SUB 340
 265 IF n$(g)(1 TO LEN t$)=t$ AN
D p=1 THEN GO SUB 345
 270 IF n$(g)(1 TO LEN t$)<>t$ T
HEN NEXT g
 275 GO TO 330
 280 CLS : FOR g=1 TO r: REM ***
****
 285 IF n$(g)(12)=c$ AND q=0 THE
N GO SUB 340: NEXT g
 290 IF n$(g)(12)=c$ AND q=1 THE
N GO SUB 345: NEXT g
 295 IF n$(g)(11 TO 12)<>c$ THEN
 NEXT g
 300 GO TO 330
 305 PRINT #2;"      DISC ";d$
 310 FOR g=1 TO r: REM **********
 315 IF n$(g)(14)=d$ AND p=0 THE
N GO SUB 340: NEXT g
 320 IF n$(g)(14)=d$ AND p=1 THE
N GO SUB 345: NEXT g
 325 IF n$(g)(14)<>d$ THEN NEXT
g
 330 GO SUB 370
 335 STOP
 340 PRINT #2;n$(g)(1 TO ): GO T
O 350
 345 LPRINT n$(g)(1 TO )
 350 BEEP .02,30
 355 RETURN
 360 CLS : PRINT "...........././.
.": PRINT n$(118)(1 TO ): REM to
 check n$()(1 TO )
 365 STOP
 370 BEEP .1,20: PRINT #0;"
      Press A Key": PAUSE 0: CLS
 : CLOSE #4: GO TO 30
 375 STOP
 380 CLS : PRINT AT 10,12;"SAVIN
G": SAVE *1;"DISCLIST3" LINE 15:
 SAVE *1;"discs3" DATA n$(): VER
IFY *1;"DISCLIST3": VERIFY *1;"d
iscs3" DATA n$(): PRINT INVERSE
1;AT 10,12;" SAVED ": BEEP .5,20
 : CLOSE #4: STOP
```

Figure 1

1 = MUSIC	M	1	9 = DRAUGHTS	P	1	17 = fill	G	1
2 = DARTS	A	1	10 = pa	P	1	18 = lister	U	1
3 = traffic	S	1	11 = <<DOOM>>	A	1	19 = T.O.H.	P	1
4 = HEXCONV	U	1	12 = util	U	1	20 = dism8kutil	U	1
5 = fruit	S	1	13 = dodgems	A	1	21 = MOON	M	1
6 = jackpot	S	1	14 = ratcatch	A	1	22 = make4	P	1
7 = bouncer	A	1	15 = life	p	1	23 = coindrop	A	1
8 = T. Kit 48k	U	1	16 = guardian	M	1	24 = drawer	G	1

Figure 2

25 =	TAB Fields	U	1	67 =	CADMASTER	G	4	109=	ARTstudio	G	6
26 =	ackack	A	1	68 =	estell	G	4	110=	super cat	U	6
27 =	headread	U	1	69 =	OXO	P	4	111=	test	M	6
28 =	stickdraw	G	1	70 =	clock	M	4	112=	mouse	S	6
29 =	Tasword(r)	M	2	71 =	sort	U	4	113=	draw	G	6
30 =	patience	S	3	72 =	TRICK	M	4	114=	ship$	G	6
31 =	SHUNTING	P	3	73 =	noise gen	M	4	115=	card$	G	6
32 =	engine	G	3	74 =	roadrunner	A	4	116=	copyprint	U	6
33 =	editkey	U	3	75 =	3d maze	P	4	117=	ironer2	G	6
34 =	toolkit	U	3	76 =	cg	M	4	118=	disclister	F	6
35 =	mysterybox	P	3	77 =	tick-tock	G	4	119=	DISASM48	U	6
36 =	BETALISTER	F	3	78 =	EXPLODE	G	4	120=	animator2	G	7
37 =	BANDIT	S	3	79 =	EXBASIC	M	4	121=	tasword	M	7
38 =	organ	M	3	80 =	mandala	S	4	122=	TLW2(last)	M	7
39 =	fade	G	3	81 =	Windowdem	G	4	123=	78recds2	F	7
40 =	DATA(beta)	F	3	82 =	betachar	M	4	124=	DISCLIST	F	7
41 =	waves	G	3	83 =	contmusic	M	4	125=	Discs4	F	7
42 =	dice	S	3	84 =	ship$	G	4	126=	discs1	F	7
43 =	circle	G	3	85 =	BETALISTER	F	4	127=	controls	M	7
44 =	Beta Basic	M	3	86 =	BETADISCLR	F	4	128=	78recds	F	7
45 =	BETADEMO	M	3	87 =	thick char	U	4	129=	write	M	8
46 =	BETADATA	F	3	88 =	futuristic	U	4	130=	ADDR	F	8
47 =	PUZZLE	P	3	89 =	factors	M	4	131=	wordmanage	M	8
48 =	brickyard	A	3	90 =	TV TEST	G	4	132=	DISCLIST	F	8
49 =	exam.mem	U	3	91 =	2xHeight	M	4	133=	discs	F	8
50 =	zoom	G	3	92 =	deleter	U	4	134=	discs2	F	8
51 =	datacalc	F	3	93 =	copyprint	U	4	135=	discs3	F	7
52 =	YS TITLE	G	3	94 =	ARCHER	A	4	136=	DISCLIST3	F	8
53 =	CLS	G	3	95 =	super cat	U	4	137=	discs3	F	8
54 =	clearscrn	G	3	96 =	Golf	S	5	138=	renumb	U	4
55 =	flash	G	3	97 =	DARTS	A	5	139=	ZZZ		
56 =	explode	G	3	98 =	Accounts	F	5	140=			
57 =	udgdef	G	3	99 =	kitchen	A	5	141=			
58 =	SOUND	M	3	100=	EFFECT	M	5	142=			
59 =	enlarger	G	3	101=	CHR.LEFT	G	5	143=			
60 =	disclister	F	3	102=	DISSEMB	U	5	144=			
61 =	alarmclock	U	3	103=	MC AID	U	5	145=			
62 =	wordsearch	P	3	104=	UDG edit	U	5	146=			
63 =	letter sq	P	3	105=	arrows	G	5	147=			
64 =	golf	S	4	106=	solitaire2	S	5	148=			
65 =	Beta Basic	M	4	107=	TABCALC	F	6	149=			
66 =	TROJAN	G	4	108=	TABDATA	F	6	150=			

which it's filed. Lines 65 to 85 direct the program to lines 100 (new entries, additions, corrections): 145 (pick any number; what's the entry?); 190 (pick entry by title, so that you can find which disc it's on, list programs, etc., of any particular category, list what is on any particular disc); 380 (SAVE routine); finally 95 (STOP).

In line 105 are Jim's titles — he says it's fatal to have a miscellaneous file! The POKEs in lines 195 and 220 are to lock and unlock caps shift; unfortunately Jim had used both capital and lower case "p" in line 105, and didn't feel like putting all the entries in again.

Lines 230 onwards contain a number of LPRINT options for which line 7 is essential if you have a printer.

Inverse print

This program therefore could form a basis for your own random access program. In this particular listing, Jim has made things more difficult for himself by inserting inverse printing directly into the listing — it looks good, but I can't easily make it show up in the printout, whereas INVERSE 1 always does (do please remember this when submitting programs). Nevertheless, the program does what it sets out to do. Perhaps a complete data base will follow. Who knows?

Errata

The program in last month's column for the modification to GENS 3 has a bug in it: the following program line will correct it. **308 IF b$(3 TO 5)="CAT" THEN CLS : CAT 1 : PRINT 0;"Any Key to Return": PAUSE 0: LET f= USR((PEEK 23670+256* PEEK 23671)−2514*(A=(PEEK 23670+256* PEEK 23671)+6291)−1857*A PEEK23670+256* PEEK 23671)+6291): GO TO 65**

CROSSWIRES

More mysteries from the Spectrum's technical Twilight Zone with Ray Elder.

Midi matters

Q *Dear Sir,*
As a new reader to your magazine I seek advice which has possibly been answered before. Being the owner of a Spectrum+ 128, having climbed the ladder from the ZX81 & Spectrum 48K, my main reason for purchasing the 128 was for the inbuilt MIDI interface.

Having a MIDI equipped synthesizer I am interested in information on the MIDI port, ie. which pin is what and what is needed so that a connecting lead can be made between the 128 and the synth. It appears that there are three connections used at the synthesizer MIDI port but I do not know what these are either.

On another note, are there any interfaces for the 128 which would enable the Commodore printer MPS801 and/or the disk drive 1514 to be used.
King regards,
A. Holmes, Exeter.

A Last thing first. To my knowledge there are no interfaces which will allow the connection of Commodore equipment to the Spectrum, Commodore went their own way and produced peripherals to their own standards and which were incompatible with just about everything else.

I would strongly recommend leaving the MIDI port alone as far as DIY is concerned, I must admit as to only having rudimentary knowledge of it, but prefer to buy ready made units instead of spending hours slaving over a hot soldering iron!

Cheetah Marketing Ltd. 1 Willowbrook Science Park, Crickhowell Road, St. Mellons, Cardiff. (Tel. 0222 777337) will sell you such a cable for £9.95. Most larger branches of Boots also stock their wares.

The MIDI port is very limited and all you are likely to be able to do is send the PLAY strings out to your synth. To do anything more interesting, such as record, arrange and sync. equipment together you need a unit with at least IN, OUT (possibly THRU) sockets and a clock or sync device.

As you have a real synth, I would strongly advise you to invest in a separate MIDI interface and software, Cheetah do a very impressive looking MIDI interface priced at £44.95.

ElectroMusic Research Ltd. of 14 Mount Close, Wickford, Essex SS11 8HG (Tel. 0702 335747) produce the PERFORMER which in my opinion is the best real time package and costs around £120, whilst XRI systems and their MICON interface have the best step time software available. Unfortunately the two are not software compatible with each other's interface. I know XRI have a real-time package but they have not yet sent us a copy to review.

XRI are at 10 Sunnybank Road, Sutton Coldfield, West Midlands (Tel. 021 382 6048) and their system costs around £108.

Mode muddles

Q *Dear Sir,*
I have a Spectrum coupled to a single Opus Discovery disc drive and since recently, a Spectrum 128K +2 which works beautifully. There is however, one problem, and that is that the sound is interfered with by a loud drone for which my supplier has no remedy.

I do not like the way in which there is no warning in 128K mode as there is in 48K mode that the computer is in CAPS LOCK or GRAPHICS modes. I was disappointed in the 128 and I gather that other readers have encountered difficulties. I congratulate Toni Baker on the article Keypad Not Required which I read with interest.
Hoping to hear from you,
W.R. Luyks, Netherlands.

A I am afraid there is nothing I can do about the lack of CAPS LOCK or GRAPHICS mode indication and the drone seems to be a mystery. I get the impression that you upgraded from a standard 48K Spectrum to the 128 +2 and I wonder if you modified the Discovery?

Opus produced an upgrade ROM to replace the existing 48K one for the discovery, I tried it out and it worked without any problem, including any unwanted sound effects. If you have not fitted the ROM upgrade then I suggest you contact Opus and do so forthwith.

If the upgrade has been done or if the Opus was supplied specifically for the 128K then I suggest you try the Opus with another 128 and if the fault persists then the fault is in the disc drive and it should be returned for checking and repair.

If there are no unwanted sounds with another Spectrum then your computer has a fault and that should be checked and repaired.

Rotronics runaround

Q *Dear Sir,*
A few months ago I purchased a DX85 printer from Rotronics of High Wycombe, Bucks. At first I could not get the machine to work using the various commands in the manual, but it worked really well when I used the software supplied with it.

Now after a little use it has stopped working and I would like it repaired. I tried to telephone Rotronics but I cannot get a reply and the GPO tell me there is nothing wrong with the line. Could you please tell me if you know whether this company has ceased trading?

Also could you tell me the address of Iamleco Computer Sys which is on a label on the back of the printer.
Thank you, yours faithfully,
D.H. Pipe, London NW10.

A I'm afraid that Rotronics are no longer trading, having recently gone into liquidation. Your best bet is to contact one of the many repair companies, of which ManComp and Video Vault are highly recommended.

Hard lines

Q *Dear Sir,*
I have recently bought an Opus disc drive for my 128K Spectrum, and whilst I am happy with its performance, I have noticed that it causes a slight interference on the TV screen.

This consists of faint horizontal lines flickering up and down the screen and, although the interference is slight, once noticed it can become maddening.

I would be grateful if you could explain the likely cause and how to get rid of it. Perhaps I have the disc drive too close to the TV?
Yours faithfully,
Philip A. Clark, Newcastle.

A You are right, this is one of those things that can drive you mad! However there is not much you can do and I suspect that it is due to one of the components of your Spectrum suffering a small, non fatal, hiccup. Sir C., in his wisdom, used components which often were only just on the operational side of the specification. Any slight change from the usual mode of operation could cause such irritating effects.

But before you give up hope, changing the position as you suggest, not only of the Opus but also of each of the other components that make up your system, in relation to each other may ease the degree of the effect and perhaps cure it.

I would also suggest that you try cleaning the edge connector at the back of the Spectrum with a cotton bud dipped in Isopropylalcohol or some "pure" clear, or using an eraser to gently remove any dirt. You can lose nothing by experimenting!

ZX COMPUTING

FULL CREDIT FACILITIES FOR ACCESS AND BARCLAYCARD.

Lineage: 48p per word. (VAT inclusive)

Semi display: £9.50 per single column centimetre + VAT. Ring for information on series bookings/ discounts (minimum 2.5cm).

All advertisements in this section must be prepaid. Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request).

01 - 437 0699

Send your requirements to:
NICOLA BATY
ASP Ltd., 1 Golden Square, London W1

ACCESSORIES

RGB OUTPUT ?
COMPOSITE VIDEO ?
GREEN SCREEN ?
TTL - ANALOGUE ?
COMP SYNC ?
BRIGHT ?

adapt electronics

CONFUSING ISN'T IT?

Talk of the problem solvers. We have the experience and expertise to advise on all aspects of computer display problems, eg:

★ **RGB OUTPUT MODULE** for Spectrum 16 or 48K. Much acclaimed — gives monitor quality pictures — no dot crawl. **£36.95**

★ **NEW** ★ **BRIGHT BOX** — Allows full use of the bright facility on analogue input monitors and monitor-TV's to produce full colour palette. Only for Spectrum 128. .... **£29.95**

Add P&P £1 (UK), £2 (EEC), £4 (Others)
Send SAE for our Free Monitor Fact Sheet

20 STARLING CLOSE, BUCKHURST HILL, ESSEX IG9 5TN. Tel: 01-504 2840

MAKE SOME SPECTRUM CeNCe

Simple Computer Controller for Machinery and Models using your Sinclair Spectrum.

A little black box which plugs into the expansion socket of any Spectrum and enables you to drive two small to medium stepper motors (up to 2 AMPs per phase) or up to 8 DC motors, relays, lamps or any other DC device.

Simple to Program and built by an engineering firm for its own use. You can use it for CNC turning, milling or coordinate drilling.

Reliable Prototype still running after three years use.

We provide Interface instructions, software, hints and examples. Price £125 inc. carriage and VAT Power supplies and Stepper Motors available. We will be pleased to quote for larger units.

P.R. PRODUCTS, PARK MILL, HOBSON STREET, MACCLESFIELD, CHESHIRE SK11 6BE.
Tel: (0625) 22800.

SPECTRUM RESET SWITCH - Prevents damage to your computer when power plug pulled in and out. **Only £3.49 inclusive**. Cheque or P.O. to Central Electronics, St Andrews St, Greenock PA15 1HG. Other accessories. Write for free list.

RING 01-437 0699 FOR DETAILS OF OUR SERIES DISCOUNTS

SERVICES

STOP PLAYING GAMES

Use your computer to make money. Turn your hobby into a home-based income. Full and part time opportunities to cash in on this tremendous market. High earnings easily possible. Open to any amateur micro user and gamer. Write for free details.

Westlink Promotions (ZX)
108 George Street
Edinburgh EH2 4LH.

REPAIRS

COMPUTER REPAIRS

We are the experts, having serviced Sinclair computers since the introduction of the ZX80.

Don't waste money on estimates – we repair Sinclair computers at price quoted (inclusive parts, labour, postage, VAT, irrespective of fault. No hidden charges.

Repairs guaranteed for 3 months.

Spectrum	£18.75 inc parts
ZX81	£11.50 inc parts
16 KRam	£9.95 inc parts
Microdrive	£15.95 inc parts
Interface 1-11	£18.75 inc parts
also	
BBC	£22.00 + parts
Electron	£19.95 + parts
XK Memory Expansion Kit	£15.95

Computer Retailers please phone for Special Trade Price.

Call or send with cheque or P.O.
T.V. Services of Cambridge Ltd.
French's Road, Cambridge, CB4 3NP
Tel. 0223 311371

SINCLAIR SERVICE CENTRE

- Fully Guaranteed Fast Repair Service
- Approx. 24hr Turnaround
- £15 av cost or send machine with £1.95 (return p&p) for free estimate.
- Personal callers welcome.

QUANTUM
33 CITY ARCADE, COVENTRY CV1 3HX
Tel: (0213) 24632

SPECTRUM REPAIRS

£14.95 inclusive of labour parts and **p&p.** Fast, reliable service by qualified engs., average repair **24hrs. 3 months guarantee**. For help or advice ring:
H. S. Computer Services, Unit 2, The Orchard, Warton, Preston, Lancashire PR4 1BE. Tel: (0772) 632686.

SCOTLAND'S No 1

For home and personal computer repairs
Specially fast Spectrum service!

- *Same day for most faults*
- *1 hour if delivered personally*
- *Open 6 days a week*
- *Free estimates*
- *Upgrades, Membranes and P.S.U.'s*
- *3 mth. warranty on work done*

Also BBC/CBN/ORIC and PERIPHERALS

MICRO-SERV
95 Deerdykes View
Westfield Industrial Area,
Cumbernauld G68 9HN
Scotland
Tel: Cumbernauld (02367) 37110
Trade, schools and club discount given.

SPECTRUM REPAIRS

From £3-£19 incl. all parts, labour and 3 month Guarantee.

Phone for special postal rates

MICRO FARE
296 Gloucester Road, Horfield, Bristol
Tel: 0272 46575

PLAN YOUR ADVERTISING BUDGET WITH ZX COMPUTING. RING 01-437 0699 NOW

Lineage rate: 48p per word (VAT inc.) Minimum **£7.20**.
Semi display: £9.50 per single column centimetre + VAT. Minimum size 2cm. No reimbursements for cancellations.
All ads must be pre-paid.
Write your advert in BLOCK CAPITALS in the grid below, ticking the section you wish it to appear under, INCLUDING YOUR NAME AND ADDRESS IN THE WORD COUNT and send it to: **ZX COMPUTING, ADVERTISEMENT DEPARTMENT, NO: 1 GOLDEN SQUARE, LONDON W1R 3AB.**

☐ REPAIRS ☐ SOFTWARE ☐ UTILITIES ☐ FOR SALE ☐ ACCESSORIES ☐ OTHERWISE PLEASE STATE

CLASSIFIED COUPON

ALL CLASSIFIED ADVERTISEMENTS MUST BE PRE-PAID. THERE ARE NO REIMBURSEMENTS FOR CANCELLATIONS.

I enclose my Cheque/Postal Order for £..........................for........ insertions, made payable to Argus Specialist Publications. (Delete as necessary)

PLEASE DEBIT MY ACCESS/BARCLAYCARD NO

£ .... FOR .... INSERTIONS EXPIRY DATE ....

Name ....

Address ....

.... POST CODE ....

DAYTIME TEL NO. ....

Signature .... Date ....

IF YOU DO NOT WISH TO CUT YOUR MAGAZINE, PHOTOCOPY THIS FORM

WARNING NOTICE

Advertisements placed in this magazine are to be in strict compliance with our standard conditions (copies of which conditions are available on request) and on the clear understanding that the advertiser warrants that his advertisement(s) does not infringe any copyright or condition of sale of any interested party in the advertised product.

Further, the advertiser indemnifies the proprietors of this magazine in respect of costs, damages, or any other claims brought against them as a result of legal action arising from the publication of the advertisement.

Any breach of these terms or the said conditions may result in prosecution of the advertiser by the proprietors.

SOFTWARE

CHEZRON SOFTWARE
(Dept. 303) 605 Loughborough Road
Birstall LEICESTER LE4 4NJ

BONABANNER — Programmers! Unleash the power of string graphics on your Spectrum. Create title pages, text adventure graphics, program headers etc. that print instantly from Basic or machine code.

At just £3.50 why hesitate! — if our demo program doesn't convince that **BONABANNER** is useful to you, JUST RETURN THE WHOLE PACKAGE FOR A FULL REFUND!!

BUDGET SPEECH synthesiser! 48K Spectrum. Just load and your Spectrum can talk through it's own loudspeaker. 100% software - No hardware needed. £3.95 from: SMB Software, PO Box 38, Inverness IVI 1GA.

RACING SOFTWARE

MAKE RACING PROFITABLE

With our top-class program. Last season 40% profit. No racing knowledge needed.
For full details and results write to:
BRIMARDON COMPUTER RACING SERVICES, 48 Pierremont Road, Darlington, DL3 6DH.

TO ADVERTISE IN OUR NEWEST SECTION RING NICOLA BATY ON 01-437 0699

PRINTER SERVICES

PRINTER OWNERS
Save Money on Printer Ribbons
Have your ribbon re-inked at less than half the price of a new one.
STANDARD PRICE ANY MAKE £1.50
Send your ribbon and £1.50 for fast service to:
K&M COMPUTERS
40 Fairstead, Birch Green, Skelmersdale, Lancs WN8 6RD.

FABRIC RIBBON RE-INKING
'Introductory Offer'
£1.50
Send ribbons + prepayment to:
ALADDINK, (Dept ZXC), 4 Hurkur Crescent, Eyemouth, Berwickshire, TD14 5AP.
Tel: 08907 50965
As seen in Crossfire Feb '87

01-437 0699 = RESULTS

UTILITIES

SPECTRUM TAPE and DRIVE TRANSFER UTILITIES

TU8 — ESSENTIAL for ALL Spectrum 48K or Plus owners — our tape utility for backup and drive conversion

* Simple operation for fast loaders — accurate even if over 48K (tone and rate measurer).
* Integrated header reader, manages headerless blocks.
* Deals with clicky loaders — converts COMPLETELY to normal for tape load (plus easy drive transfer with TPM).

Plus many other featurers — a unique product!

COST only £6.99 on tape OR £7.99 on m/drive cart.

TRANSFER PACK 2 — FOR TAPE TO DRIVE TRANSFER

Package of programs including TU8, other utilities, allows drive transfer of many programs

* Details given on transfer of 40 programs.
* Manages virtually all clicky leaders. Bytes chopper/splitter, header reader etc.

State name of drive system (e.g. m/drive, Opus disc, etc).

ONLY £11.99 on tape OR £12.99 on m/drive cart.

INFORMATION SHEETS £1 each (up to No. 18 available) includes many top 20 hits — SAE for list. OVERSEAS: add £1 Europe, £2 others for each. UPDATE SERVICE: — SAE for details.
PROGRAMS CARRY OUT MONEY BACK GUARANTEE (not updates)

LERM, DEPT ZX, 11 BEACONSFIELD CLOSE, WHITELEY BAY, TYNE AND WEAR NE25 9UW.

LETTA-HEAD PLUS

Still the most versatile 48k Spectrum utility program to design and print your own business & personal stationery: letterheads, receipts, orders, labels & posters. Create the design on screen, select the required format & print all the copies you need.

* Library of 25 different fonts including foreign alphabets.
* Fast, compiled font editor to modify and create new fonts.
* Any screen portion may be scrolled, inversed or erased.
* Box function encloses any area in a box of any thickness.
* Graphics draw, arc circle, plot, fill for logo design.
* Load & use any SCREEN$ from another program.
* Print headings in a choice of 18 sizes.
* Design labels up to 4" × 2" in size.
* Single key selects cassette label option.
* Choice of different proportionally spaced characters.
* Price £9.00

DUMPY 3.0

All the screen dumps you will ever need for your Spectrum! Unlike other screen dump programs DUMPY is a screen dump GENERATOR; it creates the machine code, relocates it, and saves it ready for you to use in your programs. No need to understand assembler, just follow the menus. The improved version now includes:

* Window dump: definable start & end of both column & line.
* Optional automatic screen area determination.
* Handles the full 24 lines of the display.
* Select from up to 7 different print densities.
* Select from up to 9 widths and 9 heights of output.
* Plain black & white or shaded colour representation.
* Tab to any position on the paper.
* Machine code can be positioned anywhere in RAM.
* Vertical or sideways dumps (for big posters).
* Price £8.50

LIN-O-TYPE

Add style to your written word; print out any word-processor file in high density NLQ in a choice of 25 fonts. Turn your Spectrum into an electronic typewriter, superb for addressing envelopes, filling in forms or writing short notes.

* Full on screen WYSIWYG line editor.
* Library of 25 different fonts including foreign alphabets.
* Up to six fonts may be mixed on each line.
* Fast, compiled font editor to modify and create new fonts.
* Single or double width characters and underlining.
* True proportional spacing and right justification.
* Load files from any storage device: mdv, tape, Wafa, disc.
* Print text as found, or edit en route to printer.
* Double density print gives NLQ with most fonts.
* Printing is multi-tasked with keyboard use.
* Price £8.50

ASTRUM+

NEW! The very best Z80 assembler and monitor package for the Spectrum. It is exceedingly versatile, powerful and easy to use and allows unlimited source code to be assembled from drive to drive. Supports microdrives or OPUS DISCOVERY.

* Full screen wordprocessor style editor.
* 256 columns wide for properly documented source code.
* Search & replace. Block copy, delete & move facilities.
* Unlimited size source code in 10k pages.
* Inbuilt macros for easy to read, compact source code.
* Assemble large programs drive to drive - including OPUS!!
* Include source files written with most other assemblers.
* Monitor features up to 9 breakpoints, single stepping etc.
* Slow run facility with 9 breakpoints in RAM or ROM (!).
* Countdown breaks after set number of passes through a loop.
* Library of useful utility programs for file handling.
* Price £13.50 (state microdrive or Opus version).

All Bradway Software programs drive Epson compatible printers (Brother, Star, M-T, Shinwa, Centronics etc.) via Hilderbay, Kempston, Tasman, Morex, Wafadrive, Interface 1, ZX LPrint III and Opus interfaces.

Prices are for cassettes & include P&P within Europe. Please add £1.50 for microdrive cartridge, £2.50 Opus disc, £2.00 worldwide airmail postage. Upgrades from previous versions (where appropriate) £2.00.

Bradway Softwre (ZX), 33. Conalan Avenue, Sheffield, S17 4PG

SOFTWARE

WD Software

FOR THE QL:

JOSS — **base £13**
Forget that tedious, time-consuming syntax! Just move the cursor and press SPACE for all your file commands. Cursor keys or your joystick allow you to access microdrives (up to 8) and floppy discs (as many as our interface allows) with up to 150 files on each! Scroll and print directories, COPY, DELETE or PRINT any file, select TV or Monitor mode before LOADing or RUNning any program. You only use the keyboard to set the DATE or label a device when FORMATting. Easy to use with Psion and other software. No silly icons to learn - JOSS will TELL you what it's going to do! Programmer's toolkit and mass copying/printing utilities also supplied. Specify microdrive-only, Microperipheral or CST-compatible disc versions.

Ref QL7 — **base £7**
1300 useful QL references with ARCHIVE 2 search/print program Too long for just one cartridge, so if you have RefQL5 just pay 2 and extra media cost to update.

Mdv Extension Cable — **£5.50**
Eight inches long, allows addition of extra microdrives to your QL. Twist it to put their slots facing you.

FOR THE QL: SPECTRUM; BBC; ELECTRON

WD Morse Tutor — **base £4**
Written to teach amateurs, now used by professionals too! Absolute beginner, or stretching your speed to 18 wpm, you won't find anything with more helpful features. What else can offer 100 random sentences as well as all the basics? Disc version unsuitable for BBC B+.

FOR THE SPECTRUM:

WorDfinder (Microdrive/disc only) — **base £8**
For CHEATING at crosswords. Finds m-ss-ng letters, solves, anagrms. 13,000 word vocabulary, so too long to share a cartridge. 10-letter word ending in ATE? No problem.

Tradewind — **base £3**
Sailing/trading strategy game with graphic surprises.

Jersey Quest — **base £3**
Text adventure in time. Background of Jersey folklore from Stone Age to Bergerac.

For export:
QL hardware and software from many sources. Ask for list/quotation.

ORDERING ADD COST OF MEDIUM, POSTAGE £1 OUTSIDE EUROPE.
Mdv or 5.25" floppy = £2 3.5" floppy = £4 Cassette = £0

Payment
By ACCESS/Eurocard/MasterCard or STERLING (UK bank cheques, Eurocheques, drafts or International GIRO). To:
WD Software (ZX), Hilltop, St. Mary, Jersery, C.I. tel: (0534) 81392

QL/SPECTRUM UTILITIES! Wide range of business/practical programs available. SAE/IRC for details (state Micro). S.D. Micro-systems (ZX) PO Box 24, Hitchin, Herts.

NEW ZX81 SOFTWARE Games, utilities, adventures, books, our ZX81 Users Club and much more. Send SAE to: A.C.V., 1 Foxwell Square, Southfields, Northampton NN3 5AT.

THE BETTER LETTER

Modifications to the Tasword character set

by John Wase and Colin Laycock

Both Tasword 2 and Tasword 3 utilize the same 64-column character set: Tasman software didn't alter it for Tasword 3; they merely changed the location. In both cases, therefore, Tasword's actual on-screen letters can be changed if one wishes: in this example many of the lower case letters have been redefined **Figure 1** to provide a character set which has much improved legibility when compared with the original Tasword letters.

Tasword 2

For Tasword 2, all you have to do is to enter BASIC after loading and do the POKES listed in **Figure 2.** To save the new character set, select the "save Tasword" option from the main menu, and the code thus saved includes the new characters.

Tasword 3

Tasword 3 is only a little less straightforward. Subtract 22784 from each of the addresses given in Figure 2. This is conveniently done by: "LET z = 22784: POKE 61706–z:, 7: POKE . . . etc.

Vive La France

All this gives you a clue as to how the Tasword letters are stored. Moreover, if you check in your Epson manual (or possibly the one for your printer if it's not an Epson), you will find that certain character codes corresponding to such things as curly brackets in the English character set correspond to accented letters in, say, the French or German set. It's then not too difficult to incorporate in Tasword 2 an addition to line 15 to soft-switch the printer to, say, the French character set; similarly a line 60 or 70 can be readily added to Tasword 3. By appropriate POKEs to the addresses for the curly bracket in the Tasword character set, you can convert this into an accented letter. Et voila! A French Tasword, both on screen and printer.

Tasword Two: Improved character set

"a":	POKE 61706,7	POKE 61708,7	
"b":	61714,7	61715,5	POKE 61718,7
"c":	61722,7	61726,7	
"d":	61730,7	61731,5	61734,7
"e":	61738,7 or 3	61740,7	61742,7
"f":	61745,6 or 2		
"g":	61754,7	61757,7	61759,7
"h":	61762,7	61763,5	
"i":	61768,2 61771,2	61769,0	61770,6
"j":	61776,1 61783,3	61777,0	61778,1
"k":	No change		
"l":	61793,6 61796,2	61794,2 61797,2	61795,2 61798,7
"m":	61805,5		
"n":	No change		
"o":	61818,7	61822,7	
"p":	61826,7	61829,7	
"q":	61834,7	61837,7	
"r":	61842,7 or 5	61843,4 or 6	
"s":	61850,7	61852,7	61854,7
"t":	61862,3		
"u":	No change		
"v":	No change		
"w":	61883,5	61886,5	
"x":	No change		
"y":	61901,7	61903,7	
"z":	No change		

Fig 2

Tasword 128

Finally, I don't know if Tasword 128 has the characters in the same addresses as Tasword 3: for the latter the set starts at base address 38144, and since this is in unpaged RAM, I would expect it to be the same for Tasword 128. Tasman assure me that Tasword 128 will shortly be put on Opus disc, and as soon as this happens, I'll find out.

a b c d e f g h i j k l m n o p q r s t u v w x y z

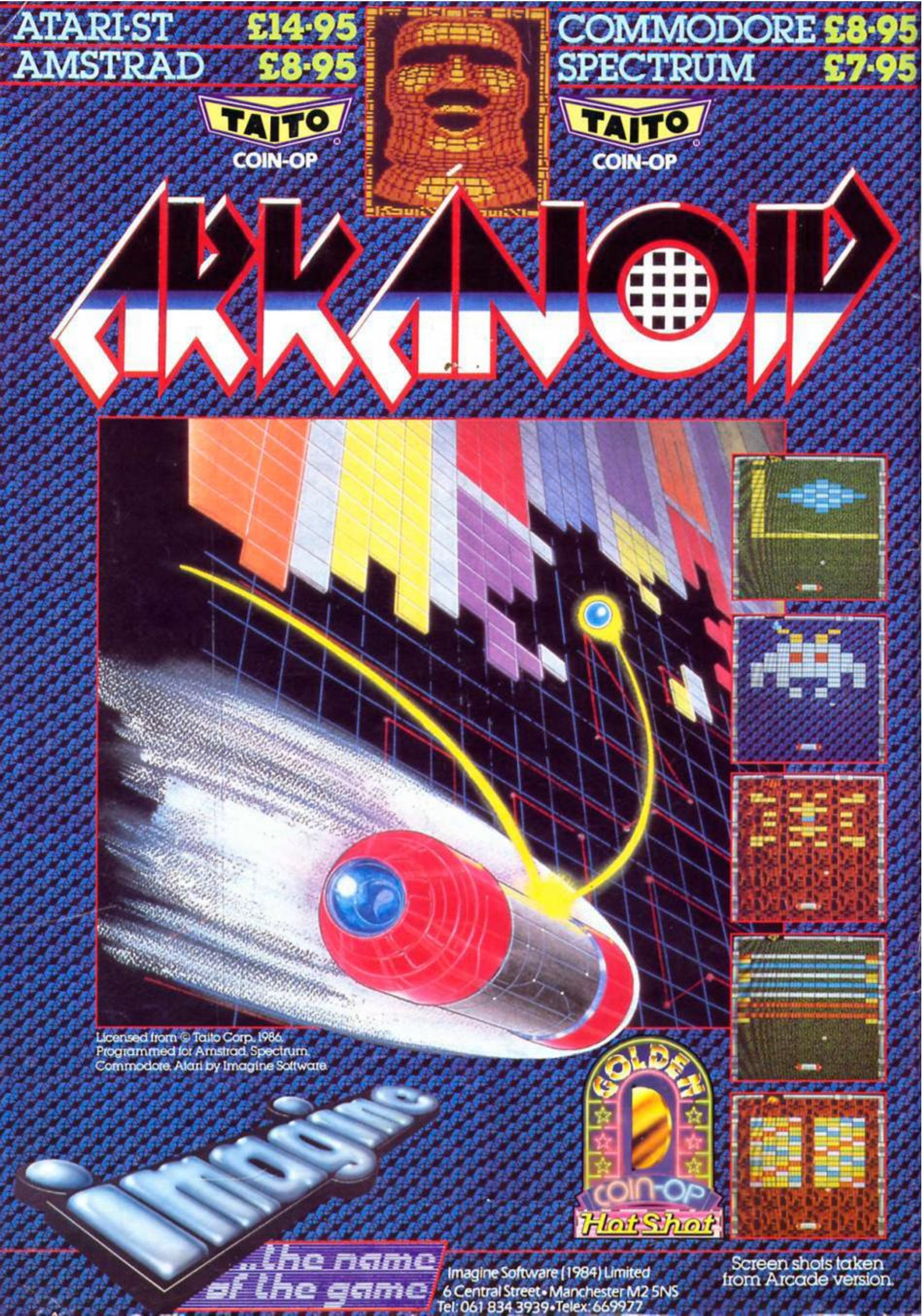
Fig 1

This is a test of the new character set for Tasword Two. Only the small letters have been changed, but there is a significant improvement in on-screen legibility.

All one has to do is to enter Basic and poke the locations with the numbers given and then SAVE the new version using option t of the main menu.

In some cases alternative POKEs are given. They should be tried and the most pleasing version selected before saving.

This character set has much improved the letters "m", "l" and "w". If you do not like the "i" and the "j" (they are now of the same height as the other letters, but merge occasionally with descenders of the line above), then do not make the POKEs for these letters.



ATARI·ST £14·95
AMSTRAD £8·95
COMMODORE £8·95
SPECTRUM £7·95
TAITO
COIN-OP
TAITO
COIN-OP
ARKANOID
Licensed from © Taito Corp. 1986.
Programmed for Amstrad, Spectrum,
Commodore, Atari by Imagine Software
GOLDEN
COIN-OP
Hot Shot
Imagine
...the name
of the game
Imagine Software (1984) Limited
6 Central Street • Manchester M2 5NS
Tel: 061 834 3939 • Telex: 669977
Screen shots taken
from Arcade version.